Common Palette Creation Algorithm for Alpha and sRGB Images

P.S.Periasamy
Department of ECE
KSR College of Engineering
TamilNadu, India

S.Athi Narayanan
Department of ECE
KSR College of Engineering
TamilNadu, India

K.Duraiswamy
Dean
K.S.Rangasamy College of Technology
TamilNadu, India

ABSTRACT
In this paper, we present a common palette creation algorithm for multiple images with transparency information. The proposed algorithm supports creation of a common palette for multiple images, transparent alpha images and flexibility to the user to add a color to the palette. This method was extensively tested for natural and synthetic images and the results are reported here. The experimental results show that the proposed method produces highest Structural Similarity Index values and outperforms existing state-of-the-art color reduction methods.

Index terms:
Color Reduction, Color Histogram, Alpha Images, Multiple Images, Human Visual System

1. INTRODUCTION
RGB color images typically can contain up to 16 million (256 ^ 3) different colors. Image display devices such as monitor and printing devices provide only a limited number of colors. So, one of the basic tasks of image processing based RTOS (Real Time Operating Systems) & DSP Processors is to reduce the number of colors in an image with minimal visual distortion. The color visualization & representation problem is aggravated when many images are displayed simultaneously, resulting in a palette size of 256 or even substantially less colors assigned to each image. Furthermore, such a coarse graining of colors is important for fast image manipulation on a reduced color palette, for computational efficiency in subsequent image processing and computer vision applications, and for image coding and transmission.

In the literature, the most popular techniques for representing a color image with a minimum set of colors are Median Cut algorithm [1], Popularity algorithm [2], and Octree algorithm [3, 4]. The median cut algorithm [1] partitions the color space into three-dimensional rectangular regions (boxes), with roughly equal numbers of pixels within each region. It is implemented by repeatedly dividing selected boxes of the space in planes perpendicular to one of the color axes. The region to be divided is chosen as the one with the most pixels, and the division is made along the largest axis. In the popularity algorithm [2] the color representatives in the color table are selected as the colors that are most populated in the image. Identify the n most popular colors, and eliminate all the other colors. This algorithm can be implemented easily by sorting the colors by pixel count. Octree [3, 4] is a data structure used to divide up the color space, while allowing fast pixel indexing with an inverse color table. Octree will be built by taking pixels, one at a time, and either making a new color or merging the pixel with an existing color. After the prescribed number of colors in the color table have been established, the new pixel either makes a new color (causing merging of two existing colors), or it is merged with an existing color. Each color is represented by an octcube, or a set of octcubes, so the merging operation effectively prunes the octree. In the ACoPa algorithm [5], hierarchical partitioning of the HSI color space is done based on the statistical color information present in the image. This method combines the K-Means clustering algorithm to create the color palette. A technique for producing scalable color quantized images is introduced in [6]. Other color image quantizing methods such as agglomerative clustering [7], sequential scalar quantization [8], kohonen neural networks [9], model based method [10], dynamic programming [11] are available in literature.

The shortcomings of the methods in the literature are as follows,

1. Creating a common palette for multiple images is not addressed.
2. Images having transparency information is not supported by the previous methods. Transparency can be present in an image, either by having a tRNS chunk [12] in an indexed color image or by having an alpha channel in a sRGB image.
3. The flexibility of adding a color by the user is not supported by the previous methods. This feature is important for images having important small objects. The pixel count of the colors present in small object will be very less compared to other colors in the image. So, the color reduction algorithm may miss out the colors present in that object, resulting in the total degradation of that object.

The proposed method solves the shortcomings and also produces better results compared to the previous methods.

The remainder of this paper is organized as follows- Section II discusses the problem considered. Section III briefs the proposed algorithm; Section IV demonstrates the GUI for add/edit colors by the user. The experimental results and their significance are summarized in Section V. Concluding remarks and future work are discussed in Section VI.

2. PROBLEM DEFINITION
When multiple images are to be displayed in one screen, the total number of unique colors present in all the images may exceed 256. For the example in Figure 1, three color images are considered, each having 256 unique colors. Assume that the union

of the 3 images has 600 unique colors. 168 colors (3x256 – 600) being repeated among the 3 images. But in a display device, a maximum of 256 colors only may be allowed. So, showing the 3 images in one screen will result in degradation of the images and the image fidelity will be lost.

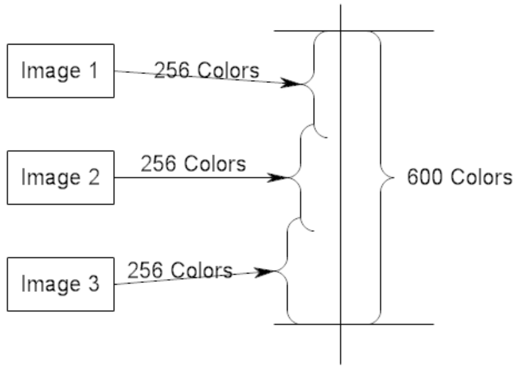


Fig. 1

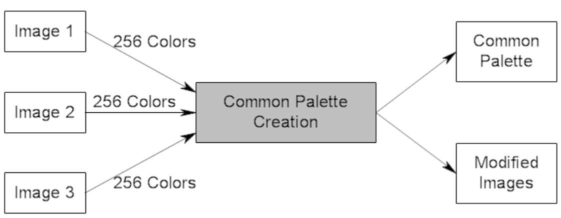


Fig. 2

As shown in Figure 2, the image processing system of the display device can attempt to create a common palette with 256 best chosen colors, so as to show the 3 images with minimum degradation. In the above example, we have considered the 3 images as indexed color images (PNG files with color type 3). The indexed color images have a maximum limit for number of colors as 256. As the number of images to be shown in one screen increases, choosing the best colors becomes tough. The scenario will become tougher when sRGB images (PNG files with color type 2) are considered, because sRGB images have no maximum limit for number of colors. Alpha images or the RGBA images (PNG files with color type 6) still worsen the problem by introducing the alpha plane to the RGB planes. The proposed algorithm is to create a common palette to show the images without degradation or with minimum degradation. The proposed algorithm creates a common color palette for a set of images to be shown in one screen.

3. PROPOSED ALGORITHM

The steps involved in the proposed algorithm are as follows:

1. Get the Input Image.
2. If the Input Image color type is not 3, convert it to color type 3.
   i. If the Input Image is of color type 2, then on conversion to color type 3, PLTE chunk will be resulted.
   ii. If the Input Image is of color type 6, then on conversion to color type 3, PLTE chunk & tRNS chunk will be resulted.
3. Remove unused colors from the color palette of each image.
   i. In an image, a color may be present in the PLTE chunk but that color may not be used anywhere in the image. So, this color unnecessarily occupies a position in PLTE chunk.

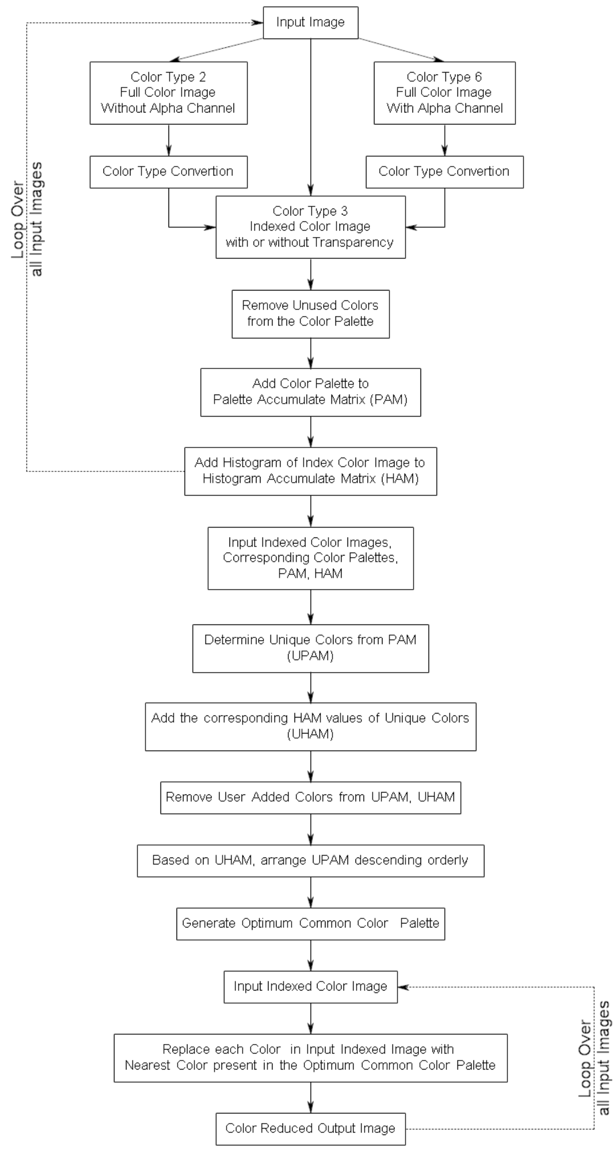


Fig. 3. Flow Diagram of the proposed algorithm

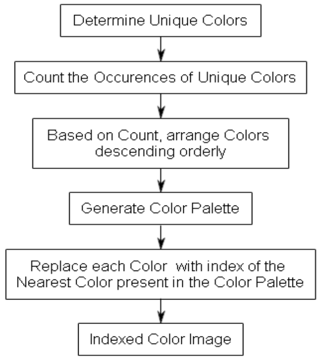


Fig. 4. Flow Diagram of the Color type Conversion

4. Add the colors present in each image to Palette Accumulate Matrix (*PAM*).
5. Add color histogram of each image to Histogram Accumulate Matrix (*HAM*).Repeat steps 1 to 5 to loop over all the images.
   i. Once the Steps 1 to 6 are completed, *PAM* and *HAM* matrices will be ready. *PAM* matrix contains all the colors present in the given set of images. *HAM* matrix contains the corresponding histogram of the colors present in *PAM* matrix.
6. Determine unique colors from the *PAM* matrix and represent it as *UPAM*.
7. Determine the corresponding histogram of the colors present in *UPAM* matrix and represent it as *UHAM*.
8. Remove the presence of user added colors from the *UPAM* and the corresponding *UHAM* values. Because these colors will definitely be present in the final common color palette.
9. Sort *UHAM* in descending order; correspondingly arrange the colors in *UPAM*.
10. With proper thresholding, generate the optimum common color palette from the sorted *UHAM* and *UPAM*.
11. For each input image from the output of Step 2, replace each color in the input Indexed color image with the nearest color present in the optimum common color palette. Nearest color will be chosen based on Sum of Absolute difference criteria.
12. Repeat step 12 to loop over all the images.
13. Thus the resultant common color palette and the corresponding color reduced images are obtained.

The details of the Color type conversion is as follows:

1. Determine the unique colors in the given image.
2. Count the occurrences of each unique color and divide by the size of the image.
3. Based on the count, arrange the colors in descending order.
4. With proper thresholding, generate the color palette with a maximum of 256 colors.
5. Replace each color in the input image with the nearest color present in the color palette.
6. Thus the resultant indexed color image with PLTE chunk is obtained.

3.1 Global Thresholding

In the proposed method, global thresholding is used for color type conversion of an image. If the number of colors in the image is greater than 256, then we will check the count value of the 257^{th} color. If it is high means, the color distribution of the image is flat. In this case, we will set a threshold; so that, the distance (sum of absolute difference) between any two colors in the generated color palette will be greater than the threshold. Thus the generated color palette can cover the entire color distribution of the image. The threshold should be directly proportional to the count value of the 257^{th} color.

3.2 Adaptive Thresholding

In the proposed method, adaptive thresholding is used for common palette creation for multiple images. Let K be the number of colors to be present in the common palette. A high value in the K^{th} element of the *UHAM* matrix means that, for the given set of images the color distribution is very diverse. The threshold for a particular color in *UPAM* is defined as:

$$T_i = \left(\frac{UHAM_{\max} - UHAM_i}{UHAM_{\max} - UHAM_{\min}} \right) \text{X } T_{\max}$$

where T_{max} represents the maximum threshold, $UHAM_{max}$ is the maximum value in the *UHAM* matrix, $UHAM_{min}$ is the minimum value in the *UHAM* matrix, $UHAM_i$ is the UHAM value of the i^{th} color present in the *UPAM* matrix. i^{th} color will be present in the common palette, if the distance (sum of absolute difference) between the i^{th} color and each of the colors already present in the common palette is greater than T_i.

A color will have high *UHAM* value, if the probability of occurrence of that color is high in the given set of images. According to the above equation, a high *UHAM* value will generate a low threshold, thus the probability of occurrence of that color in the common palette will be high. So that, the degradation of the set of images after color reduction will be minimum. Similarly, a color will have low *UHAM* value, if the probability of occurrence of that color is less in the given set of images. According to the above equation, a low *UHAM* value will generate a high threshold, thus the probability of occurrence of that color in the common palette will be less. So that, infrequent colors will not occupy the common color palette.

4. GUI FOR ADD/EDIT COLORS BY USERS

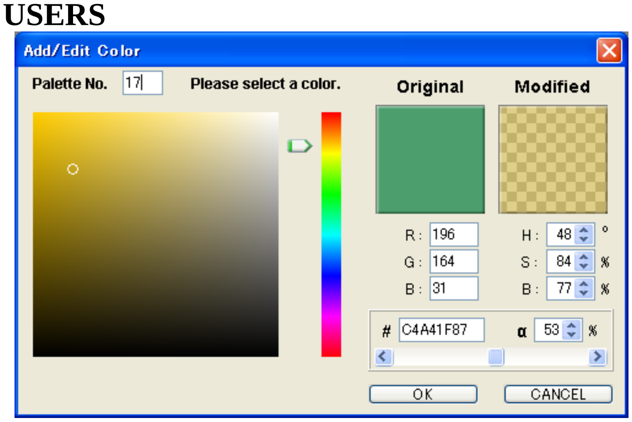


Fig. 5 GUI for add/edit colors

The User is having the flexibility to add/edit colors of the common palette. Before performing color reduction of images, user can add a color to the common palette. For example, in a set of images, if text information is present in the images, the text color can be added to the common palette prior to color reduction. Any such highly significant color can be added by the user prior to color reduction. After the common palette is created, user can edit any of the colors in the common palette and perform color reduction again, so as to view the impact of edited color on the images. In advertising program creation for Digital Televisions, it is common that, in a screen, a set of images & background color of the advertisement will be fixed and another set of images will be changed frequently. In such scenarios, a *base common palette* (with minimum number of colors) can be created with the fixed images by performing color reduction. This *base common palette* can be considered as reserved "*user added colors*" for the subsequent processes. The remaining empty cells in the palette can be used by the next color reduction process for the other set of frequently changeable images. Thus the base common palette can be reused for the next new set of frequently changeable images in the advertisement program. The GUI for add/edit colors by the users is shown in Figure 5.

5. EXPERIMENTAL RESULTS

The proposed algorithm was designed in Matlab 6.5 and the application was developed in Java 1.5. The proposed algorithm has been tested with a large set of natural and synthetic images. The image set includes

1. sRGB images.
2. Images with Alpha channel.
3. Indexed color images
    i. With 256 colors in PLTE Chunk
    ii. With less than 256 colors in PLTE Chunk
    iii. With tRNS chunk (With Transparency)
    iv. Without tRNS chunk (Without Transparency)

5.1 Quality Metrics

The MSE (Mean Square Error) is not well-correlated with the human visual system. So, we cannot use RMSE (Root Mean Square Error) or PSNR (Peak Signal to Noise Ratio) as a quality metric for comparing color reduced images with the original images. We need a quality metric which mimics the human visual system. For evaluating our method, we have used the Structural Similarity Index value (SSIM) described in [13] with default parameter settings. The SSIM is a method for measuring the similarity between two images. The SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data. A high (low) SSIM index value means good (poor) quality. For a color reduced image, the mean SSIM (MSSIM) index value is calculated over R, G & B planes by overlapping windowing operation and the average value of the three is taken as the quality metric for comparison.

5.1.1 Natural Image results

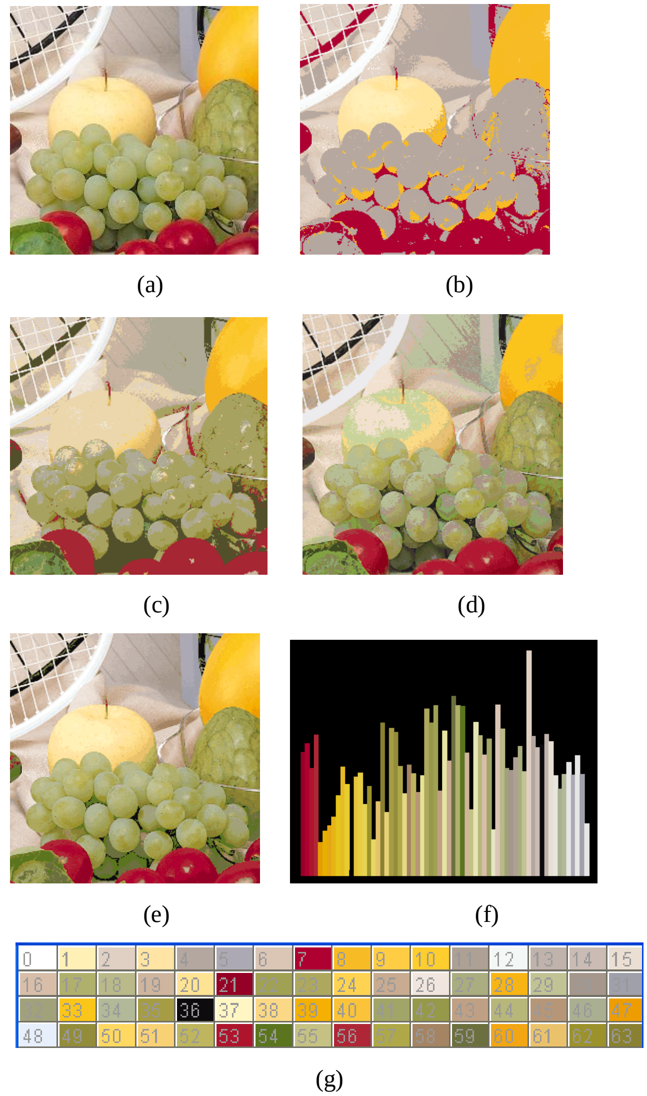
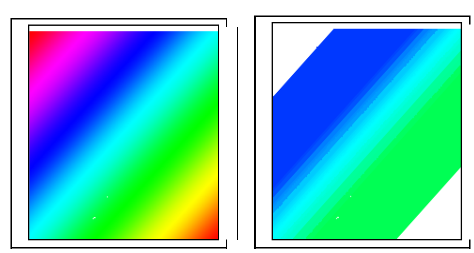


Fig. 6 (a) Original Image, (b) Popularity algorithm with 64 colors, (c) Octree algorithm with 64 colors, (d) Median-Cut with 64 colors, (e) Proposed method with 64 colors, (f) Histogram of figure (e), (g) The corresponding palette of the figure (e).

Figure 6 shows the results of different methods of color reduction for the fruits image. Original image is a 256 X 256 sRGB image with 22,756 unique colors. The popularity algorithm and the Octree quantization produces contouring effect on the color reduced image. In the Median-Cut algorithm result, color mis-mapping is visible. The color reduced image of the proposed method looks significantly good for the human visual system

5.1.2 Synthetic image results



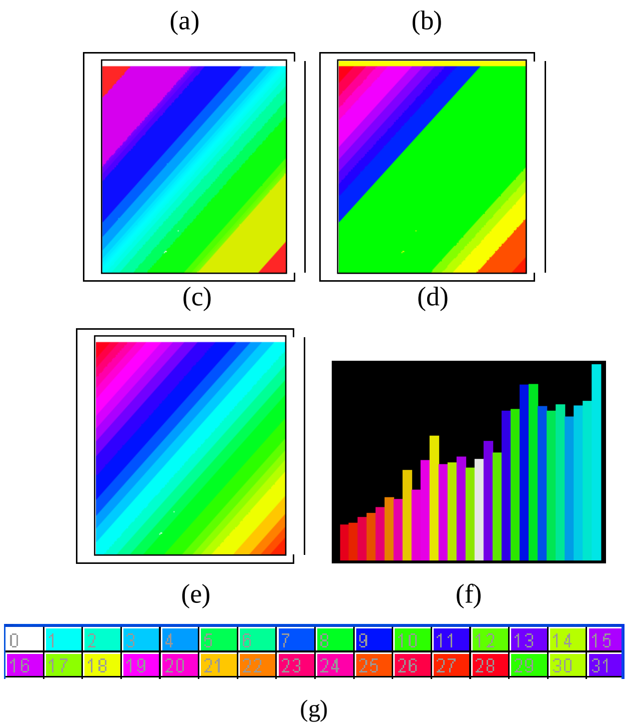


Fig. 7 (a) Original Image, (b) Popularity algorithm with 32 colors, (c) Octree algorithm with 32 colors, (d) Median-Cut with 32 colors, (e) Proposed method with 32 colors, (f) Histogram of figure (e), (g) The corresponding palette of the figure (e).

Figure 7 shows the results of different methods of color reduction for the synthetic RGB image. Original image is a 205 X 200 sRGB image with 773 unique colors. The results of the proposed method are superior to that of the Popularity algorithm, Octree quantization and Median-Cut algorithm.

Table 1. Average SSIM values of natural image

Image	Natural Image			
No. of Colors	Popularity	Octree	Median-Cut	Proposed
16	0.51659	0.66302	0.62562	0.69953
32	0.57634	0.68341	0.69948	0.79327
48	0.62318	0.73022	0.74196	0.83383
64	0.62635	0.7323	0.75891	0.89169
80	0.63264	0.77043	0.77596	0.90364
96	0.68553	0.7718	0.81069	0.91133
112	0.70152	0.783	0.81413	0.91645
128	0.70529	0.83504	0.82467	0.92155
144	0.70884	0.85537	0.82884	0.92307
160	0.71968	0.86371	0.83735	0.92486
176	0.73727	0.86451	0.84356	0.92632
192	0.74194	0.87786	0.85496	0.92676
208	0.74335	0.87658	0.85582	0.92703
224	0.74403	0.88436	0.85936	0.92727
240	0.7446	0.88756	0.86073	0.92735
256	0.75634	0.89048	0.86089	0.9276

Table 2. Average SSIM values of synthetic image

Image	Synthetic Image			
No. of Colors	Popularity	Octree	Median-Cut	Proposed
16	0.64551	0.64243	0.61079	0.81079
32	0.67252	0.74009	0.65847	0.82318
48	0.69809	0.81317	0.67049	0.89896
64	0.83047	0.83546	0.68507	0.9012
80	0.86066	0.88213	0.68854	0.95096
96	0.90265	0.90522	0.69414	0.9595
112	0.9119	0.93453	0.69886	0.97377
128	0.91621	0.94463	0.71781	0.98145
144	0.9271	0.95459	0.71869	0.98515
160	0.94819	0.96651	0.71929	0.98663
176	0.95437	0.97124	0.71943	0.98673
192	0.96183	0.97555	0.72022	0.98983
208	0.96796	0.97845	0.72065	0.99147
224	0.97263	0.98164	0.72149	0.99275
240	0.97489	0.984	0.72265	0.99412
256	0.97733	0.98574	0.73764	0.99515

Table 1 and 2 shows the Average SSIM values for the Fruits image and synthetic RGB image respectively by different methods versus the number of colors in the color reduced image. It's clear from Figure 8 that, the proposed method produced the highest SSIM values for all the cases.

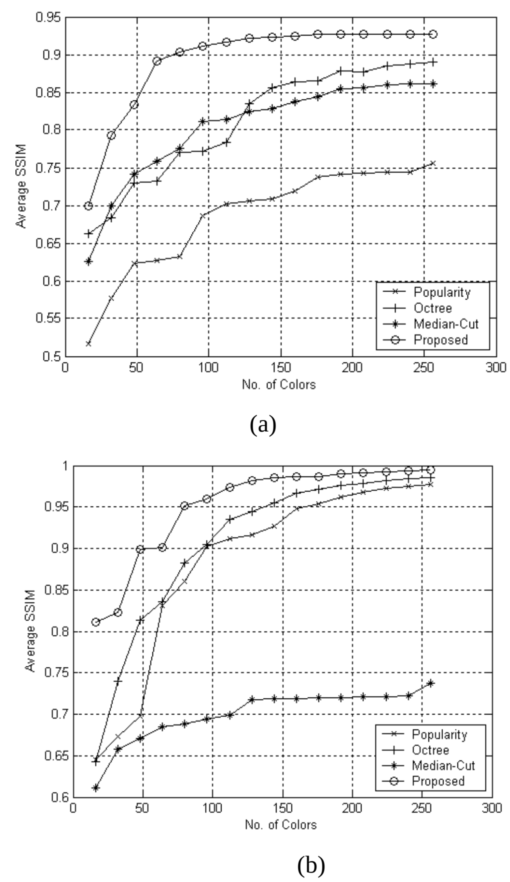


Fig. 8 (a) Average SSIM values plot for the natural image, (b) Average SSIM values plot for the synthetic image

5.1.3 Multiple sRGB images

sRGB images (*true color images*) have 3 planes, R, G & B. The introduction of an additional plane results in the possibility of increased number of colors in the image. sRGB color images typically can contain up to 16 million different colors. In the literature, displaying multiple images in a screen involves only fixed. Creating a common palette for a given set of images is not addressed in the literature. Proposed method is able to create common palette for multiple images. Simulation results for sRGB images are shown in Figure 9. All the 5 images are color reduced to have a common color palette of 256 colors.

Table 3. Alpha Channel Image results

	Input		Color Reduced		
Image	Unique Color count	File Size (in Kb)	Unique Color count	File Size (in Kb)	Average SSIM
Image 1	14650	43.8	256	11.6	0.9002
Image 2	8027	50.8		11.8	0.9527
Image 3	7670	47.2		10.8	0.9597
Image 4	13313	59.0		16.2	0.9815
Image 5	7581	43.4		9.0	0.9272

The 5 sRGB input images shown in Figure 9 have 51,241 unique colors but we are representing these images with just 256 colors. Table 3 depicts the significant file size reduction and color reduction using the proposed method.

5.1.4 Multiple Images with Alpha Channel

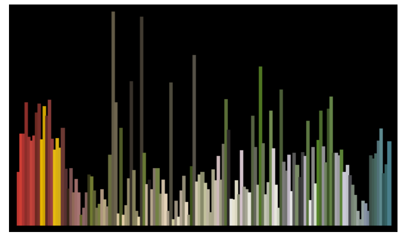
Alpha channel images have 4 planes, R, G, B & Alpha. The introduction of an additional plane results in the possibility of increased number of colors in the image. RGBA color images typically can contain up to 4 billion (256 ^ 4) different colors. Displaying multiple images with alpha channel is not addressed in the literature. Proposed method is able to create common palette for multiple images with alpha channel. Simulation results for alpha channel images are shown in Figure 11. “House” image has been kept as background in order to illustrate the alpha channel (transparency supported) images. In order to visualize the transparency in each color, background checker board pattern is used in Figure 11 (c). The five Alpha channel images shown in figure 11 have 73,389 unique colors but we are representing these images with just 256 colors. Table 4 depicts the significant file size reduction and color reduction using the proposed method.

Table 4. Alpha Channel Image results

	Input		Color Reduced	
Image	Unique Color count	File Size in Kb	Unique Color count	File Size in Kb
1-1 Alpha Color Grid Test	95	5.5	256	2.3
2-1 Alpha Pattern Test	8979	41.2		9.4
2-2 Alpha Pattern Test	39383	99.0		17.9
3-1_Alpha Mat Test	9602	32.8		7.6
4-1 Multiple Alpha Test	15330	39.3		7.3

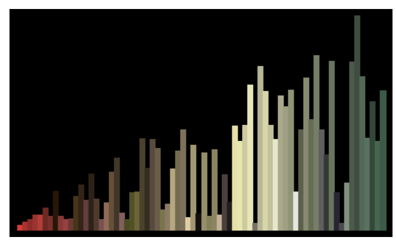

(a)


(b)


(c)

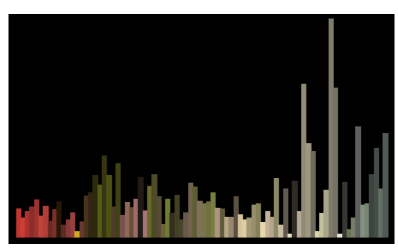

(a)


(b)

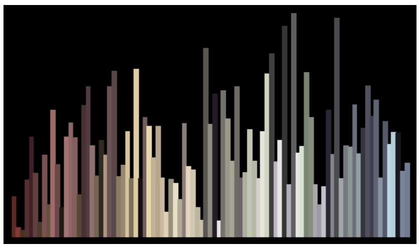

(c)


(a)


(b)

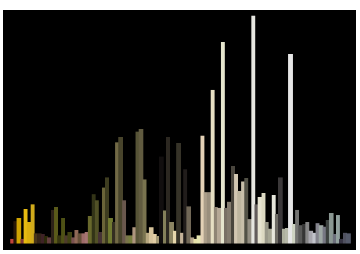

(c)

(a) **(b)**



(c)

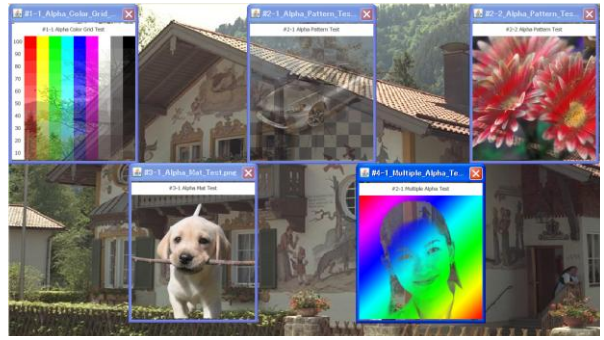
(a) **(b)**



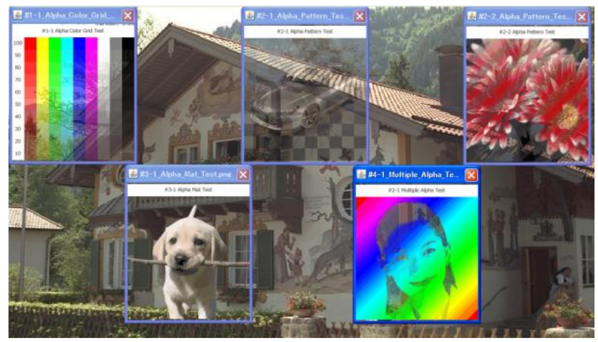
(c)

Fig. 9: (a) Original Images, (b) Color reduced Images using proposed method, (c) Histogram of Middle Images.

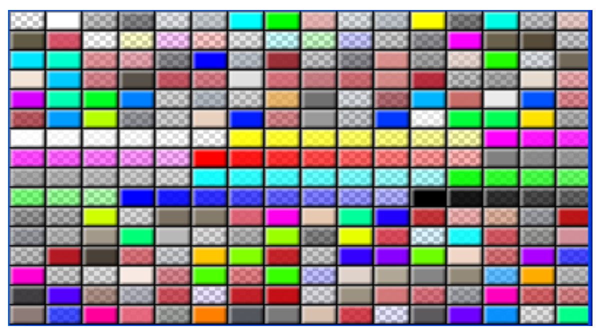
Fig. 10: Common color palette of color reduced images in Fig. 9.



(a)



(b)



(c)

Fig. 11 (a) 5 Alpha Channel Images before color reduction with House Image as Background, (b) Corresponding color reduced images with 256 colors, (c) Corresponding common color palette.

6. CONCLUSION AND FUTURE WORK

We have presented a novel color reduction algorithm which supports creation of a common palette for multiple images, transparent alpha images and flexibility to the user to add a color to the palette. This method was extensively tested with synthetic and natural images and the results are reported here. The experimental results show that the proposed method produces excellent results and outperforms existing state-of-the-art color reduction methods. The proposed algorithm can be adapted for digital broadcasting applications. The obtained palette can be combined with a dictionary of color names in order to provide a qualitative image description. Such qualitative image description can be used for image retrieval from large image databases. As the proposed method gives significant data reduction of the color information, either this method can be considered as a pre-processing module for color image segmentation or spatial information can be incorporated into the proposed method and direct segmentation results can be achieved. . The proposed method can be used to create optimum thumbnail view images for Web design. Representing a set of images with a common palette significantly increases the redundancy of pixels in an image and thus reduces the bandwidth required to transmit the images over web. This feature can be incorporated with image coding and transmission.

7. REFERENCES

[1] A. Kruger, 1994, “Median-cut color quantization,” Dr. Dobb’s Journal, pp. 46-54 and 91-92.

[2] D. Clark, 1995, “The popularity algorithm,” Dr. Dobb’s Journal, pp. 121-127.

[3] D. Clark, 1996, “Color quantization using octrees,” Dr. Dobb’s Journal, pp. 54-57 and 102-104.

[4] M. Gervautz and W. Purgathofer, 1990, “A simple method for color quantization: octree quantization,” in A.Glassner, ed, Graphics Gems I, Acad. Press, pp. 287-293.

[5] J. Delon, A. Desolneux, J.L. Lisani, A.B. Petro, 2007, “Automatic Color Palette,” Inverse Problems and Imaging, vol. 1, no. 2, pp. 265–287.

[6] Yik-Hing Fung, Yuk-Hee Chan, 2006, “A Technique for Producing Scalable Color-Quantized Images With Error Diffusion,” IEEE Trans. Image Processing, vol. 15, no. 10, pp. 3218–3224.

[7] Z. G. Xiang and G. Joy, 1994, “Color image quantization by agglomerative clustering,” Comput. Graph. Applicat., vol. 14, no. 3, pp. 44–48.

[8] R. Balasubramanian, C. A. Bouman, and J. P. Allebach, 1994, “Sequential scalar quantization of color images,” J. Electron. Imag., vol. 3, pp. 45–59.

[9] A. Dekker, 1994, “Kohonen neural networks for optimal color quantization,” Network: Computation in Neural Systems, vol. 5, pp. 351–367.

[10] T. J. Flohr, B. W. Kolpatzik, R. Balasubramanian, D. A. Carrara, C. A. Bouman, and J. P. Allebach, 1993, “Model based color image quantization,” Proc. SPIE, vol. 1913, pp. 265–270.

[11] X. Wu, 1992, “Color quantization by dynamic programming and principal analysis,” ACM Trans. Graph., vol. 11, no. 4, pp. 384–372.

[12] Portable Network Graphics (PNG) Specification (Second Edition), http://www.w3.org/TR/PNG/

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, 2004, "Image quality assessment: From error visibility to structural similarity," IEEE Transactios on Image Processing, vol. 13, no. 4, pp. 600-612.