A method for performing color reduction in index images

Hsiu-Hui Lee and Shih-Chieh Lo
Department of Computer Science and Information Engineering
National Taiwan University
hh_lee@csie.ntu.edu.tw, p91922004@ntu.edu.tw

***Abstract**-In this paper, the transcoding proxy server is provided to dynamically convert the image content to cover the difference between various capabilities of mobile devices. To fulfill the requirement of higher performance for performing transcoding at real time, we propose a method with better performance and lower operating cost to reduce the color amount in palette of an index image.*

Keywords: *Color reduction, Image transcoding, Palette, Index color*.

1. Introduction

Full color digital displays typically use 24 bits to specify the color of each pixel on the screen [7]. The memory size of the display card decides the dimension could be displayed. "Color Quantization" was proposed [3] to resolve the cost issue. This method limits the number of bits used to specify the color of each pixel. Usually, 8 bits are provided which means that only 256 colors can be simultaneously displayed. A table, called palette, is then used to define a mapping from the full range of 2^{24} colors to the 2^8 indices. But the performance and the computational cost are major issues for color quantization [5]. Many variance methods are proposed and tried to improve the performance [4] [6]. With the enhancement of hardware, PCs are almost supporting true color. But mobile devices meet same problem as PCs met before.

Various mobile devices are with different capabilities, such as supporting file format, file size or image dimension restriction. Image transcoding server [2] is proposed to resolve the problems caused by the compatibilities between various mobile devices. To perform image transcoding for different capabilities of various mobile devices, the tasks are about file format converting, dimension or data type matching, and file size compressing. Since the real time transforming is necessary in such kind of server, the performance and cost saving issue is very critical. For this reason, some tasks with performance issue before have to enhance, or they will not perform in the transcoding server.

This research will propose a method to reduce the color amount in palette with good performance. Performing color reduction is usually for making the image file size smaller. Since the less components used, the less possible patterns will occur. And it will help to enhance the compression effect [1].

2. System Overview

The architecture of the image transcoding system is illustrated in figure 1. This system consists of three primary components: the content analyzer, the policy engine, and the transcoding unit.

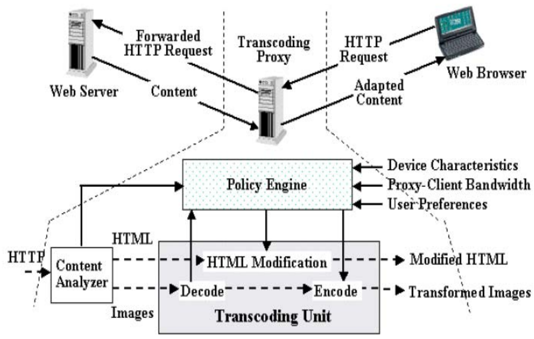

Figure 1. The transcoding proxy system

The method we proposed in this paper will enhance the performance of the transcoding unit in the transcoding system. Our method will be a sub-routine of the system that has to do image transcoding tasks. In such kind of system, they have to convert image to other form in order to fit the criteria of other kinds of devices. The architecture of the transcoding unit is shown in figure 2.

Our method takes the palette and the raw data of an index image as input parameter. After applying the reduction method, it will generate a new palette with less color entries than original palette. A mapping table is also generated for converting the index value stored in original raw data to a new index value mapped to the new palette.

In image transcoding system, it must have several routines to perform different kinds of transcoding tasks. The image read/write engines are also necessary to read the image raw data from different image formats as input parameter for the transcoding routines. The performed result can be encoded to image file again by the image libraries.

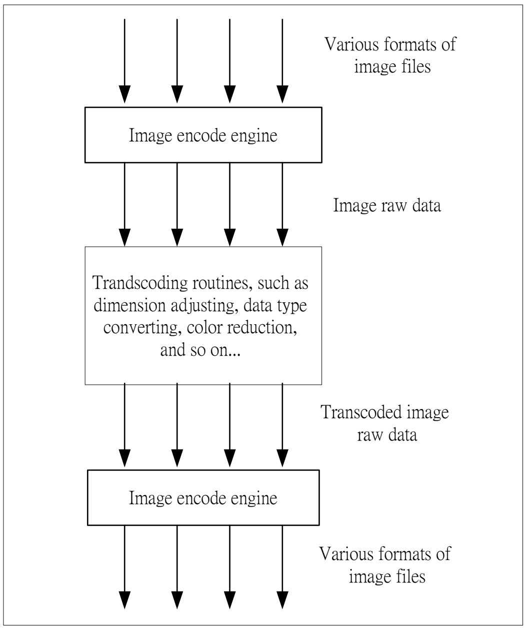

Figure 2. The working model in transcoding unit

3.Working Flow

Our palette reduction method comprises in following pseudo code:

READ *org_palette*
READ *raw_data*
READ *threshold_value*
INIT *mapping_table*
INIT *res_palette*
COMPUTE the histogram of the *raw_data*
OBTAIN the order_table
FOR each *index1* from 0 to sizeof(*org_palette*) -1
  FOR each *index2* from 0 to sizeof(*res_palette*) -1
    COMPUTE *color_diff* between
      org_palette[order_table[index1]] and
      res_palette[index2]
  END FOR
  IF all *color_diffs* are larger than *threshold_value*
    ADD *org_palette[order_table[index1]]* into
      res_palette
    OBTAIN *new_index* as the index of new added
      color.
    SET *mapping_table[index1]* as *new_index*
  ELSE
    OBTAIN *mapped_index* as the index of the
      color in *res_palette* with lowest *color_diff*
    SET *mapping_table[index1]* as *mapped _index*
  END IF
END FOR
FOR *index3* from 0 to sizeof(*raw_data*)-1
  SET *raw_data[index3]* as
    mapping_table[raw_data[index3]]
END FOR

Table 1. Notation table

Notation	Description
FCr	Red value of the first color.
FCg	Green value of the first color.
FCb	Blue value of the first color.
SCr	Red value of the second color.
SCg	Green value of the second color.
SCb	Blue value of the second color.

The formula to calculate color difference:

$(FCr - SCr)^2 + (FCg - SCg)^2 + (FCb - SCb)^2$

The smaller the compared value is, the more similar these two colors are. Here is an example to present how our method performs the palette reduction. Figure 3 shows the raw data and the palette entries of the example image. The dimension of the image is 14 * 6. The palette has 7 entries. The Threshold is given as 15.

Original image raw data

0	1	1	2	3	3	4	4	4	5	5	6	6	6
0	1	1	2	3	3	4	4	4	5	5	6	6	6
0	1	2	2	3	3	4	4	5	5	5	6	6	6
0	1	2	2	3	3	4	4	5	5	5	6	6	6
0	1	2	2	3	3	4	4	5	5	5	6	6	6
0	1	2	2	3	3	4	4	5	5	5	6	6	6

Palette entries

0	(11, 11, 11)
1	(13, 13, 13)
2	(7, 7, 7)
3	(9, 9 ,9)
4	(2, 2, 2)
5	(6, 6, 6)
6	(0, 0, 0)

Figure 3. Palette and raw data of source image

The initial tasks include building a order table regarding the histogram and a mapping table. Figure 4 shows the initial state.

Step 1:Checking the first entry of the order table, and getting the index 6, (0, 0, 0). Picking the color in the first palette. Since the second palette is empty, add the color in the second palette directly. Assigning 0 to the entry with index 6 in the mapping table. Refer Figure 5.

Step 2:Picking next color by referring next entry of the order table, and getting the index 5, color (6, 6, 6). After the calculating:
1. with SC_0: $(6-0)^2 + (6-0)^2 + (6-0)^2 = 108 > 15$

Adding this color into the second palette. Assigning 1 to the entry with index 5 in the mapping table. Refer figure 6.

Step 3:Picking next color index 4, color (2, 2, 2). After the calculating:
1. with SC_0: $(2-0)^2 + (2-0)^2 + (2-0)^2 = 12 <= 15$
2. with SC_1: $(2-6)^2 + (2-6)^2 + (2-6)^2 = 48 > 15$

Mapping this color to the index 0, (0, 0, 0), in the second palette. Assigning 0 to the entry with index 4 in the mapping table. Refer Figure 7.

Step 4:Picking next color index 3, color (9, 9, 9). After the calculating:
1. with SC_0: $(9-0)^2 + (9-0)^2 + (9-0)^2 = 243 > 15$
2. with SC_1: $(9-6)^2 + (9-6)^2 + (9-6)^2 = 27 > 15$

Adding this color into the second palette. Assign 2 to the entry with index 3 in the mapping table. Refer Figure 8.

Index distribution statistic in original image raw data

Index	0	1	2	3	4	5	6
Count	6	8	10	12	14	16	18

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	-1	Empty
Index 1	(13, 13, 13)	5	-1	
Index 2	(7, 7, 7)	4	-1	
Index 3	(9, 9 ,9)	3	-1	
Index 4	(2, 2, 2)	2	-1	
Index 5	(6, 6, 6)	1	-1	
Index 6	(0, 0, 0)	0	-1	

Figure 4. Initial state

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	-1	(0, 0, 0)
Index 1	(13, 13, 13)	5	-1	
Index 2	(7, 7, 7)	4	-1	
Index 3	(9, 9 ,9)	3	-1	
Index 4	(2, 2, 2)	2	-1	
Index 5	(6, 6, 6)	1	-1	
Index 6	(0, 0, 0)	0	0	

Figure 5. Result of step 1

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	-1	(0, 0, 0)
Index 1	(13, 13, 13)	5	-1	(6, 6, 6)
Index 2	(7, 7, 7)	4	-1	
Index 3	(9, 9 ,9)	3	-1	
Index 4	(2, 2, 2)	2	-1	
Index 5	(6, 6, 6)	1	1	
Index 6	(0, 0, 0)	0	0	

Figure 6. Result of step 2

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	-1	(0, 0, 0)
Index 1	(13, 13, 13)	5	-1	(6, 6, 6)
Index 2	(7, 7, 7)	4	-1	
Index 3	(9, 9 ,9)	3	-1	
Index 4	(2, 2, 2)	2	0	
Index 5	(6, 6, 6)	1	1	
Index 6	(0, 0, 0)	0	0	

Figure 7. Result of step 3

Step 5:Picking next color index 2, color (7, 7, 7). After the calculating:

1. with SC_0: $(7\text{-}0)^2 + (7\text{-}0)^2 + (7\text{-}0)^2 = 147 > 15$
2. with SC_1: $(7\text{-}6)^2 + (7\text{-}6)^2 + (7\text{-}6)^2 = 3 <= 15$
3. with SC_2: $(7\text{-}9)^2 + (7\text{-}9)^2 + (7\text{-}9)^2 = 12 <= 15$

Mapping this color to the index 1, (6, 6, 6), in the second palette. Assigning 1 to the entry with index 2 in the mapping table. Refer Figure 9.

Step 6:Picking next color index 1, color (13, 13, 13). After the calculating:

1. with SC_0: $(13\text{-}0)^2 + (13\text{-}0)^2 + (13\text{-}0)^2 = 507 > 15$
2. with SC_1: $(13\text{-}6)^2 + (13\text{-}6)^2 + (13\text{-}6)^2 = 147 > 15$
3. with SC_2: $(13\text{-}9)^2 + (13\text{-}9)^2 + (13\text{-}9)^2 = 48 >15$

Adding this color into the second palette. Assigning 3 to the entry with index 1 in the mapping table. Refer Figure 10.

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	-1	(0, 0, 0)
Index 1	(13, 13, 13)	5	-1	(6, 6, 6)
Index 2	(7, 7, 7)	4	-1	(9, 9, 9)
Index 3	(9, 9 ,9)	3	2	
Index 4	(2, 2, 2)	2	0	
Index 5	(6, 6, 6)	1	1	
Index 6	(0, 0, 0)	0	0	

Figure 8. Result of step 4

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	-1	(0, 0, 0)
Index 1	(13, 13, 13)	5	-1	(6, 6, 6)
Index 2	(7, 7, 7)	4	1	(9, 9, 9)
Index 3	(9, 9 ,9)	3	2	
Index 4	(2, 2, 2)	2	0	
Index 5	(6, 6, 6)	1	1	
Index 6	(0, 0, 0)	0	0	

Figure 9. Result of step 5

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	-1	(0, 0, 0)
Index 1	(13, 13, 13)	5	3	(6, 6, 6)
Index 2	(7, 7, 7)	4	1	(9, 9, 9)
Index 3	(9, 9 ,9)	3	2	(13, 13, 13)
Index 4	(2, 2, 2)	2	0	
Index 5	(6, 6, 6)	1	1	
Index 6	(0, 0, 0)	0	0	

Figure 10. Result of step 6

Step 7:Picking next color index 0, color (11, 11, 11). After the calculating:

1. with SC_0: $(11\text{-}0)^2 + (11\text{-}0)^2 + (11\text{-}0)^2 = 363 > 15$
2. with SC_1: $(11\text{-}6)^2 + (11\text{-}6)^2 + (11\text{-}6)^2 = 147 > 15$
3. with SC_2: $(11\text{-}9)^2 + (11\text{-}9)^2 + (11\text{-}9)^2 = 12 <= 15$
4. with SC_3:$(11\text{-}13)^2 + (11\text{-}13)^2 + (11\text{-}13)^2 = 12 <= 15$

Mapping this color to the index 2, (9, 9, 9), in the second palette. Picking (9, 9, 9) is because it has lower index number in the second palette. It also means color (9, 9, 9) has higher using rate in original image raw data. Assigning 1 to the entry with index 2 in the mapping table. Refer Figure 11.

Step 8:New palette and the mapping table are generated. Applying the mapping table to convert the original raw data to the new raw data of the image. Refer Figure 12.

	First Palette	Order table	Mapping table	Second Palette
Index 0	(11, 11, 11)	6	2	(0, 0, 0)
Index 1	(13, 13, 13)	5	3	(6, 6, 6)
Index 2	(7, 7, 7)	4	1	(9, 9, 9)
Index 3	(9, 9 ,9)	3	2	(13, 13, 13)
Index 4	(2, 2, 2)	2	0	
Index 5	(6, 6, 6)	1	1	
Index 6	(0, 0, 0)	0	0	

Figure 11. New palette and mapping table

New image raw data

2	3	3	1	2	2	0	0	0	1	1	0	0	0
2	3	3	1	2	2	0	0	0	1	1	0	0	0
2	3	1	1	2	2	0	0	1	1	1	0	0	0
2	3	1	1	2	2	0	0	1	1	1	0	0	0
2	3	1	1	2	2	0	0	1	1	1	0	0	0
2	3	1	1	2	2	0	0	1	1	1	0	0	0

Index distribution statistic in new image raw data

Index	0	1	2	3
Count	32	26	18	8

Figure 12. New raw data and the distribution

4. Experimental results

Regarding the palette analysis method, we implemented a program with C++ language. This program can decode a Gif image file to get the palette and the raw data of the image. Then it will follow the analysis and mapping steps to perform the palette color reduction. At last, the program will encode the new palette and raw data to a new Gif file.

In our experiment, we perform the test with 2321 image files. Most of the source files are generated from digital camera. The test source files includes various kinds of contents, such as portrait, landscape, indoor, outdoor, night scene, natural scene, artificial drawing…etc. For each testing image, we apply the palette reduction process with incremental threshold values. The threshold value starts from 10 and plus 5 each time.

The variance of the palette size and the compression rate is related with the image content itself. When the original palette colors of the image are very similar, the color reduction and the compression effect are strong with small threshold value. Regarding the testing experiment, in most cases the compression rate can be lower than 50 % with the threshold value 25. The quality of image is always relative with how many colors can be utilized in the image. In more of testing case, the quality of the result image is unsatisfied with the threshold value more than 70, unless the contrast of the image is high. On the other side, the compression rate is usually less than 20% with the threshold value more than 60. We made the statistics for both result palette size and the compression rate. Please refer Figure 13 and figure 14.

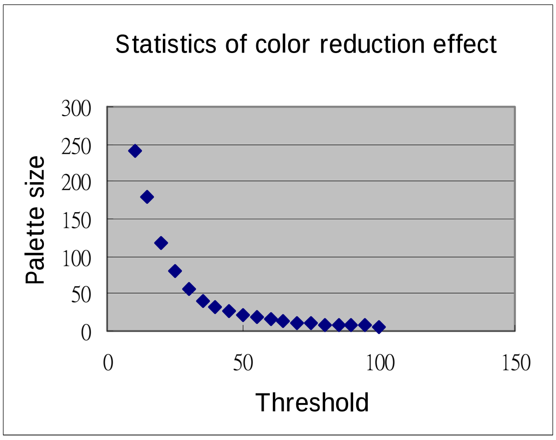

Figure 13. Statistics of color reduction effect

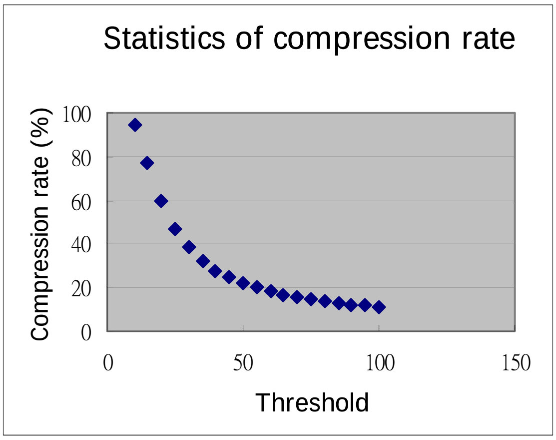

Figure 14. Statistics of compression rate

One typical result set with the threshold value, new palette color amount, and the encoded Gif file size is listed in Table 2. The profiles of the testing image are following: file size: 2337k, dimension: 2304 * 1728, palette color amount: 256. Figure 15 is the original testing image.

Obviously, the color amount in the new palette is less and less with the increasing of the threshold value. The encoded Gif file size is also smaller and smaller. In this sample, threshold 35 reaches 50 % compression rate, refer figure 16. Threshold 70 reaches the 25 % compress rate and reduces the color from 256 to 16, refer figure 17. When the threshold value is higher than 70, the existing colors in the palette are less than 16. Although the color reduction can be keep executing, the quality of the result image will be un-satisfied with limited colors.

Table 2. The results with various threshold values

Threshold value	New palette colors	Gif file size (KB)	Compress rate
10	256	2337	0 %
15	235	2211	94.6 %
20	152	1803	77.2 %
25	128	1617	69.2 %
30	104	1447	61.9 %
35	70	1189	50.9 %
40	56	1068	45.7 %
45	44	882	37.7 %
50	37	798	34.1 %
55	30	697	29.8 %
60	23	605	25.9 %
65	20	598	25.6 %
70	16	589	25.2 %
75	16	566	24.3 %
80	14	523	22.4 %
85	12	450	19.3 %
90	12	458	19.6 %
95	10	414	17.7 %
100	9	414	17.7 %

Figure 15 Original testing image

Figure 16 Testing result with threshold value 35

Figure 17 Testing result with threshold value 70

The approach we used to measure the performance is setting the time stamp in the testing program. In our experiment, we tried to proof that the performance of this method is stable. On the other words, the performance measure value should keep fitting in a certain range regardless of the source images. Regarding the testing results, the performance values always stay in a fixed range. The exact performing value will change with the hardware environment. In our usual testing hardware environment, the average performing value is under 5 milliseconds.

The testing sample images include variable kinds of image with different dimensions, palettes, and raw image contents. In the palette reduction method we used, it focuses on the palette of each image. Our method analysis the colors in the palette, re-generate a new palette, and re-map original image raw data to the new palette. Since the color count of the palette in an index image is limited, always less than or equal to 256, the computing effort used in the process of analysis palette is also limited. Other part of whole process is to remap the image raw data to the new palette. Although the dimension of the image will impact the computing effort taken in the process, the remapping is a low computing power needed process. In the real testing, the dimension of the image does not impact the performance obviously.

5. Conclusion and Further Works

5.1. Conclusion

In conventional palette reduction methods, the given parameter is usually the color number of the new palette. With color quantization, it can generate a new palette with precise number of colors. The primary drawback of color quantization is the performance issue. The other disadvantage is the dimension of the source image will impact the process time and the resource usage. The main reason for the shortcomings of color quantization applied for palette reduction is because it performs analysis with the content raw data of image.

In our proposed method, it simply focuses on the colors in the original palette. According to the histogram of each color used in the image, the method performs the palette color analysis and generates a new palette. It also creates a table for mapping the index numbers stored in original image raw data to new index numbers referring with the new palette. The primary advantage of the method is the performance. Since it only analysis the palette colors, the performance is much better than performing color quantization.

The other benefit of the method is for the case that several image frames share one palette. Some file formats, like Gif and Tiff, support multiple image frames existing in one image. These kinds of images usually only prepare one palette for all image frames to save the storage. If performing color quantization to reduce palette for multiple-frame images, it will perform the analysis for each frame and generate new palette for each frame. It will break the original situation that all frames share one palette, and increase the storage usage. Meanwhile, the performance will be a big issue, since each frame will perform color quantization. Our approach will totally resolve this problem. With our method, it can just analysis the original palette with the first frame, or the most important frame, to generate a new palette and a mapping table. Then apply the re-mapping process in each frame with the mapping table. The new image will still remain sharing one palette. The most important is the performance will be much better.

Generally, the proposed method in this paper will provide good performance and also solve some problems. Regarding the benefits, this method can be adopted in the real time or some heavy-using environment, like transcoding server, to fulfill their requirement about the performance.

5.2. Further Works

The parameter of the palette reduction method is a threshold value. The drawback of the method is that it is hard to reduce the palette color to a precise number. In current implementation, it only based on the given threshold value to perform palette reduction. In the future, we plan to build a prediction mechanism regarding the original palette and the expected palette color number to predict the given threshold value.

The other direction is to enhance the quality of the colors in the new palette. Currently, we only remain some existing colors to replace other similar colors to reduce the color number. Next step, we can modify the color picking approach to adopt the color weight concept. The general idea is generating a new color from several similar colors regarding their used weight in the image raw data rather than simply using one to replace other similar colors.

References

[1] S. Battiato, G. Gallo, G. Impoco, F. Stanco, "A colour reindexing algorithm for lossless compression of digital images", *Computer Graphics*, pp.25-28, April, 2001.

[2] A.Gaddah, A. El-Shentenawy, T. Kunz, R. Hafez, "Image transcoding proxy for mobile Internet access", IEEE 56th, vol. 2, pp.24-28, September, 2002.

[3] M.Y. Orchard, C.A. Bouman, "Color quantization of images", *IEEE Transactions*, vol. 39, Issue 12, pp. 2677-2690, December, 1991.

[4] E. Rendon, L. Salgado, J.M. Menendez, N. Garcia, "Adaptive palette de termination for color images based on Kohonen networks" *Image Processing Proceedings*, 1997.

[5] M. Semenov, "A fast algorithm for choosing the proper palette in real image representation", Image Processing conference, pp.530-533, April, 1992.

[6] Y. Wu; C. Yang, T. Wang, "An new approach of color quantization of image based on neural network", *Signal Processing Proceedings,* 2000.

[7] A. Zaccarin, B. Liu, "A novel approach for coding color quantized images" *IEEE Transactions,* vol. 2, pp. 442-453, October, 1993.