Colorization-based Coding By Using Watershed Segmentation For Optimization

Wang Ping[*], Lee Byung Gook [*]
[*]Dept. of Visual Contents, Dongseo University

Abstract
Colorization is a method using computer to add color to a black and white image automatically. The input is a grayscale image and some representative pixels (RPs). The RPs contain the color information for the image, and it indicates each region's color information. Colorization-based coding is a novel way for lossy image compression, it decodes a color image to get grayscale image and extracts RPs from the image. Because RPs decides the region's color and we also want small data size for image compression, form this viewpoint the paper proposes a way to get better and fewer RPs based on watershed segmentation. According to the segmentation result we also improve the original chrominance blending colorization method to save decode time and get better reconstruct image.

1. Introduction

As the multimedia and Internet technologies are widely used in our daily life, people can get information from images and videos instead of tedious characters. But the original image always with a large data size, it will take a long time for transmission, lots of image compression algorithms such as JPEG are used to reduce data size then the image can transfer in a short time. Nowadays a novel way called colorization-based coding for image compression appears in our sight, at first the measure decodes a color image and converts the image from RGB color space into YCbCr color space to get the luminance component (a grayscale image), then extract the RPs from the chrominance components, at last combine them together as a new image file used for transmission. As the computer received the image, use colorization method as a decoder to reconstruct the image. The transmission data is a grayscale image and some RPs' information.

2. Previous Work

The colorization-based coding for color image compression includes two parts one is colorization and the other is extracting pixels from the original image.

2.1 Colorization Use Chrominance blending

In 2006, Liron *et al*. introduce a colorization algorithm using the chrominance blending [1]. They give the colorization assumption: if there is a change in luminance it also has a related change in chrominance, sharp luminance means an edge in the chrominance, and graduate luminance

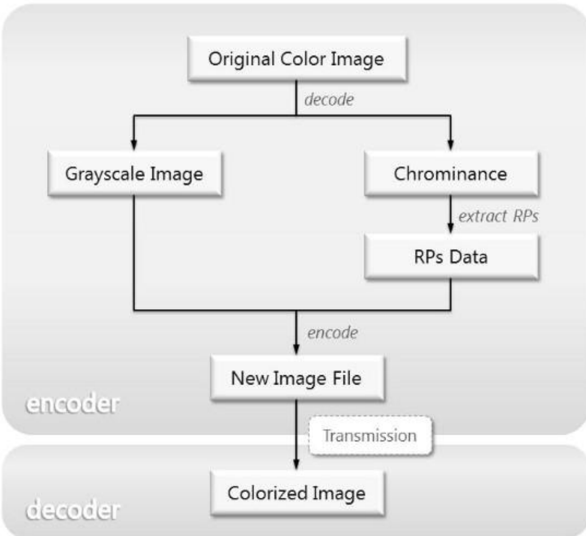

Figure 1.Overview of image compression by using colorization-based coding

means not having an edge but a moderate change. So they use minima intrinsic distance to weigh the influence from the RPs (P_s) to the pixel which should to be colorized (P_t). If the intrinsic distance is short that means P_t and P_s will have the similar color, otherwise they will have different colors. The intrinsic distance between two nearby pixels defines as

$$d(P_x, P_y) = |Y_{P_x} - Y_{P_y}| \tag{1}$$

In (1), P_x and P_y are two pixels, Y_{P_x} and Y_{P_y} are for each luminance value. It's also called the link structure. The intrinsic distance from P_s to P_t consists of some link structures, and the link structures will have amount of connections. They use the Dijkstra's algorithm to find a minima one.

$$d(P_s, P_t) = min\{\textstyle\sum_s^{t-1} d(P_s, P_{s+1})\} \tag{2}$$

After that calculate P_t chrominance value:

$$c_t = \frac{\sum_i c_i w\left(d_{c_i}(t)\right)}{\sum_i w\left(d_{c_i}(t)\right)} \quad (3)$$

$$W(d) = d^{-B} \quad (4)$$

In (3), c_t is to be colorized pixel's chrominance value, c_i is the RPs' chrominance value. In (4) B is the blending factor and it typically being 4.

2.2 Extract RPs Method

An efficient way to choose the RPs is necessary, if we just choose the RPs randomly, some small region will not have any RPs. Chen et al. [2] and Takamichi Miyata et al. [3] proposed their ways to extract the RPs. In our method, we use image segmentation to do color clustering, and then extract RPs from each cluster to insure we can get all color.

3. Proposed Method

In our proposed method, we use the grayscale image to do the watershed segmentation first, and then extract the RPs from each segmented regions. We also optimize the colorization method based on the segmentation.

3.1 Watershed Segmentation

Watershed segmentation [4][5] is a way of automatically separating similar luminance regions from a gradient image. In section 2.1, we assume that if the luminance is similar and the chrominance will also be similar. So in each region we can choose a few pixels to represents for whole region's chrominance information.

3.2 Extract RPs from Segmentation result

A Matlab in built watershed function was used to do segmentation, but it has a problem called over segmentation, to solve this problem the smooth filter should be used before segmentation. After segmentation we can get regions' labeled by index. For each region, choose the start pixel as RP, and compare with follows if one pixel's chrominance changed much, we set the pixel to be the new beginning and let it to be the RP.

3.3 Improvement of Colorization

By using watershed segmentation we can optimize the colorization method.

1. Use a threshold to the minima intrinsic distance

$$d(P_s, P_t) = \min\{\textstyle\sum_s^{t-1} d(P_s, P_{s+1})\} < \text{threshold}$$

If the intrinsic distance is very long from P_s to P_t that means they are in different color clusters, P_s and P_t have different chrominance, so we don't need to consider about it anymore.

2. Each RP works in the region they belong to.

We get RPs from the watershed segmentation regions, so we just let it effects in that region. It can save the decoding time.

4. Experimental Result


Figure2. Original image


Figure3. Grayscale image

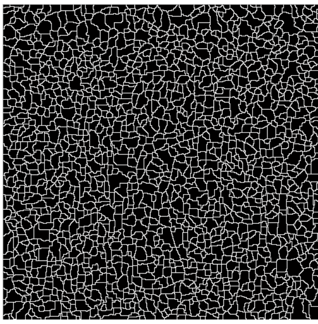
Figure4.Segmented image

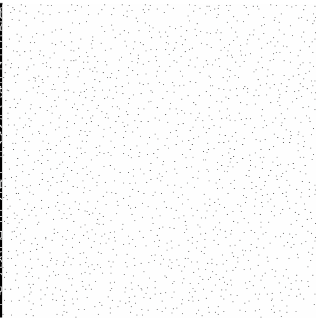
Figure.5 RPs image


Figure6. Result image

Fig2. is the original 448*448 bmp image, the size is 588KB. Fig3.is the grayscale image, the size is 54.1KB, saved as JPEG format. Fig4. is the segmented image by using watershed transform. In Fig5. the black points represent for the RPs. The number of RPs is 1379, almost 0.68% of the original image pixels, and it's better than the method mentioned in [2].The data of each RP can be coded as a four dimensional vector, two for the coordinates and two for the chrominance components. It takes 8 Byte for every RP [6].So the data size by using our compression method will be 54.1KB+ 8*1379/1024Byte=64.8KB.

We use the peak signal-to-noise ratio (PSNR) method to compare our result with the JPEG algorithm. The JPEG images from different levels were getting from Photoshop. Then calculate the PSNR with the original bmp image. The bigger PSNR value means better quality image.

Table 1. Compare with the JPEG method

Image	PSNR(dB)	Data size(KB)
JPEG Lv1	24.91	59.9
JPEG Lv2	26.61	67.7
OUR METHOD	27.71	64.8
JPEG Lv3	31.70	93.6

JPEG Lv4	43.29	162.0

Comparing row 3 and row 4 in the table, the JPEG in level 2 the PSNR is 26.61db, our method is 27.72dB, and the JPEG image size is 67.7KB, our method is 64.8KB. It indicates that our method have better quality and smaller image data size.

5. Conclusion

In the experiment our method has a better performance than the JPEG algorithm under some condition, but it still has some problems. The first one is the calculating time. Our method will take a long time than JPEG, even though I already optimize the colorization algorithm. Second one is we still have the over segmentation problem, the image is divided into too much regions. As we get the pixel from the region, large number of regions will increase the number of RPs. In [7] the author proposes an improve method to do the watershed segmentation. I will import their method as next step to improve our algorithm.

Reference

[1]Liron Yatziv, Guillermo Sapiro, "Fast Image and Video Colorization Using Chrominance Blending,"IEEE transactions on image processing. Vol.15, NO.5, MAY 2006

[2]Li Cheng, S.V.N. Vishmanathan, "Learning to Compress Image and Videos," Proceeding of the 24th International Conference on Machine Learning, Corvallis, OR, 2007

[3]Takamichi Miyata,Yuuki Komiyama, Yoshinori Sakai, "Novel Inverse Colorization For Image Compression," IEEE, Picture Coding Symposium, May 2009.

[4]Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing," pp. 622-626, 2002.

[5]Rafael C. Gonzalez, "Digital Image Processing Using Matlab," pp. 417-425, 2009.

[6]Megumi Nishi,Takahiko Horiuchi, and Hiroaki Kotera, "A Novel Picture Coding Using Colorization Technique," NIP21: International Conference on Digital Printing Technologies Baltimore, MD; pp. 380-383,September 2005.

[7]Vivek George Jacob, Sumana Gupta, "Colorization of Grayscale Images and Videos Using A Semi-automatic Approach," IEEE, ICIP,2009