An Automatic Image and Video Colorization Algorithm based on Pattern Continuity

Zhong Zhen,qquad Gui Yan,qquad Ma Lizhuang
Shanghai Jiao Tong University
zizi_19@sjtu.edu.cn

Abstract

This paper proposes a novel method to colorize grayscale images from a reference image. Rather than depending on the comparisons between the grayscale image and the reference image based on the luminance value and neighborhood statistics of each pixel, our method consider pattern continuity and spatial consistency. We handle the pixels in the grayscale image from a higher level and gain the global results, which are better than the results based on the local statistical information. Another advantage of our method is we need less human intervention than the previous methods. This is very beneficial for the video colorization.

Our method is focus on colorizing cartoon images and videos. With regard to the cartoon images, using texture analysis based on pattern continuity can gain a better result. For cartoon videos, our method generates scribbles automatically instead of drawing by the user. Once a representative frame has been colorized, all the other frames will be colorized sequentially, which makes a great improvement on automatic colorization.

1. Introduction

Colorization is a computer-assisted process of adding color to a monochrome image or video. It is a term introduced by Wilson Markle in 1970 to describe the process for adding color to black and white movies [1]. Now the term ”colorization” generally represents the process of adding color information to monochrome still images and videos. The traditional method for still images involves delineate region boundary carefully to segment images into regions and choose color from palette for each region. Unfortunately, the automatic segmentation algorithms fail to identify the complex boundaries, even manually segmentation is very difficult. For videos, it requires tracking regions across video frames, what is worse, the existing tracking algorithm is not robust to track non-rigid regions and produce intolerable errors. Thus, the traditional method is time-consuming and labor-intensive. In recent years, many colorization techniques have been proposed, which greatly reduced the amount of user efforts.

To colorize a grayscale image or video, our aim is to get the chrominance information. There are commonly two ways. The first method is depending on the scribbles provided by the user. The user specifies which parts of the image should be colorized by which color [2]. This method can get high quality result, but drawing scribbles for a complex image is challenging.The second method is providing a reference image that has similar content or style with the input image [3]. This method need little interactive, but the result is heavily depending on the quality of the reference image.

In this paper, we propose a new colorization algorithm for images and videos, which takes the advantages of the previous methods and tries to avoid their shortages. Our method needs a partially segmented reference image. For each pixel in the reference image, we construct a pattern feature vector using the Gabor wavelet transform coefficients. Then we get a classifier based on the training set. For the pixels of the grayscale image, we classify the pixels and gain the color information from the corresponding region. Finally, we preserve the more reliable pixels as the scribbles and employ the optimization-based method to get the final result.

This paper makes several specific technical contributions. First, the process of colorization is almost automatic, require little user effort. Once the reference image is provided, all the other input grayscale image with similar style can be colorized automatically. Next, our method generate the scribbles spontaneously which is more important for video colorization, as drawing scribbles manually needs a great amount of work. Furthermore, we use the pattern feature vector as the training set, which is more sophisticated, monochrome texture descriptors, accordingly generate better result. Our method is much more beneficial for video colorization; for we can colorize one frame flawlessly and all the rest frames can be colorized automatically using our algorithm.

2. Previous Work

To facilitate the colorization of images and videos, numerous methods have been proposed. Welsh *et al.* [3] proposed a semi-automatic colorization algorithm by transferring colors from a reference image to a grayscale image. They examine the luminance values in the neighborhood of each pixel in the input image and transfer the color from pixels with matching neighborhoods in the reference image. This method produces ideal results, but it fails to colorize the pixels that have similar luminance while have different colors. To solve the problem, the user must specify the swatches between the two images to direct the transfer process. Their idea is inspired by a method of color transfer between images [4].

Irony *et al.* [5] presented a novel method to colorize grayscale images by transferring color from a segmented reference image. They use a supervised learning technique to classify feature space and account for each pixel from a higher-level angle to enforce spatial consistency.

Sykora *et al.* [6] proposed a new framework for black-and-white cartoon video. The dynamic part of the scene is represented by a set of outlined homogeneous regions which superimpose the static background. They combined unsupervised image segmentation, background reconstruction, and structural prediction to reduce manual intervention.

These methods use the reference image as their chromatic information provider. Another class of colorization algorithms is user-guided methods.

Levin *et al.* [2] introduced a simple yet effective method. The user scribbles desired colors in the interiors of the regions, and then it automatically propagates the colors to the entire image by solving a simple constraint quadratic optimization problem.

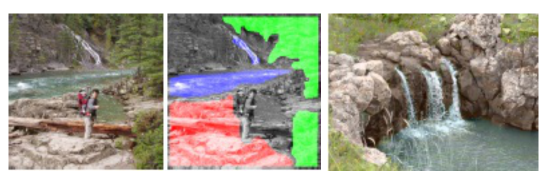
Yatziv *et al.* [7] proposed a faster scribble-based color optimization technique by chrominance blending. They colorize pixels with a weighted average of stroke colors, where the weights are proportional to the geodesic distances between the pixel and corresponding stroke. A geodesic distance measures the variation of luminance values along a path between two pixels.

In general, the scribble-based methods work better than the color transfer methods. But they require intensive user intervention, especially when the input image contains complex patterns or structures. Thus, Qu *et al.* [8] proposed a novel colorization method over regions exhibiting pattern-continuity as well as intensity-continuity. They analyze the texture space and define an affinity to measure distances between a pixel and a stroke in feature space. Then for a given stroke they evolve a level set around it to associate a spatially-coherent region with that scribble. This method is designed to deal with manga illustrations which can be successfully segmented into regions of homogeneous textures. Luan *et al.* [9] employed texture continuity to colorize natural images and also get excellent results.

3. System Overview

Our algorithm is inspired by the work of Irony *et al.* and Qu *et al.*. Irony *et al.* [5] presented a new technique for colorizing grayscale images by example. This method uses a supervised learning technique to classify the feature vector and a voting technique to emphasize the spatial consistency. Rely on the example image, this method can generate the scribbles spontaneously and colorized the input image automatically. Compared with the method of Levin *et al.* [2], this method needs less human intervention, which can be seen from the Fig. 1, saving substantial time and labor. On the other hand, this method uses DCT coefficients to analyze the texture. Since DCT analysis is too simple as a texture descriptor, the classification results of this method is defective, which narrows the scope of this method.

(a) Levin *et al.*'s method. Left: grayscale image with scribbles Right: colorization result



(b) Irony *et al.*'s method. Left: reference image Middle: segmented reference image Right: colorization result

Figure 1. Comparisons between Levin *et al.*'s method and Irony *et al.*'s method.

Qu *et al.* [8] proposed a new colorization method for manga images. It uses Gabor wavelet for texture analy-

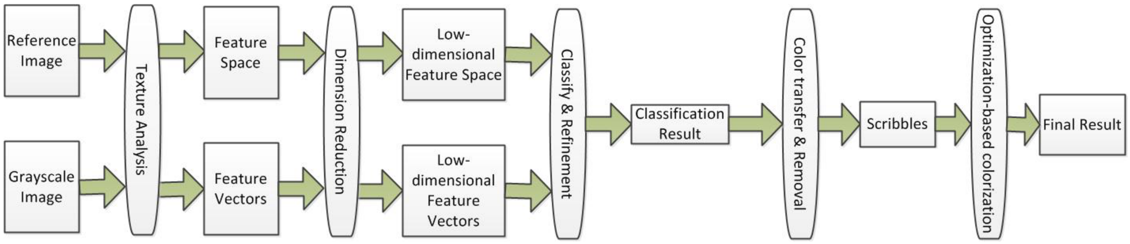


Figure 2. Overview of our framework

sis that is better than simple DCT analysis, consequently generates better classification result. However, like the method of Levin *et al.* [2], this method is laborious and tedious as well.

For video colorization, the simplest method is to colorize the videos frame by frame, which needs vast user interactive, so it is not a good option. Nowadays the most popular method for video colorization is optical flow. It can generate impressive results for some videos, but because of its poor tolerance, for many videos, it will produce huge error that cannot be accepted. Especially for the scribble-based colorization method, the automatic transmissions of scribbles are not very good, which produces bad effects. While, our approach is particularly applicable for video colorization, for it produces scribbles automatically. Compared with the traditional methods which require a large quantity of labor and time, our approach is competitive for less interactive. To colorize a video clips, we can extracted a representative frame, and all the rest frames will be colorized without human intervention.

From the above, this paper introduces a new technique for image and video colorization that requires less manual intervention, in the meanwhile, uses Gabor wavelet to obtain a better result. The framework of our method is shown in Fig. 2.

4. Algorithm

Our algorithm colorizes a grayscale image or video based on the reference image, which should have similar content or style with the input image. The reference image should be segmented into several different regions, each of which is assigned a unique class number. The segmentation can be implemented automatically using the existing segmentation algorithms, or can be manually marked by the user. For simplicity, we segment the reference image manually. Figure 3 shows the inputs of our framework: the reference image along with its segmentation and the input grayscale image. For every pixel in the reference image, we use Gabor wavelet analysis to compute the feature vector, forming the training set. Then we practice a learning technique to get a classifier. Given the pixel to be colorized in the input image, we classify the pixel and assign a class number to it. We refine the classification result considering spatial consistency. On the basis of the classification result, we get the color from the reference image according to the neighborhood information. Finally we keep the more reliable pixels as the scribbles and apply the optimization-based method to obtain the final colorization result. Figure 2 shows the steps of our method, it mainly contains two sections: classification and colorization.

(a) reference image (b) segmentation (c) grayscale image

Figure 3. The inputs of our framework

4.1 Classification

Once we get the segmented reference image, our first task is to form the training set and the classifier. Because the classifier is applied to the input grayscale image, so the training set must not depend on the color information. The best choice to solve this problem is to adopt the feature vector based on texture. Irony *et al.* choose the DCT coefficients of a k by k neighborhood around the pixel as its feature vector. The DCT coefficients are simple texture descriptor, and they are not sensitive to translations and rotations. But they are not applicable to the images with complex textures, which are commonly seen in the animation images and videos. For these images and videos, the classification result using DCT coefficients is poor. For this reason, we use Gabor wavelet to analyze patterns. The Gabor wavelet is particularly appropriate for texture representation and discrimination since its frequency and orientation are similar to those of the human visual system.

Gabor function is given as:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \quad (1)$$

Its Fourier transform G(*u,v*):

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \quad (2)$$

Where $\sigma_u = \frac{1}{2}\pi\sigma_x$ and $\sigma_v = \frac{1}{2}\pi\sigma_y$.

Then a cluster of self-similar filter dictionary $g_{m,n}$ can be acquired by scaling and rotating the coefficients g(x,y) :

$$g_{m,n}(x,y) = a^{-m}G(x', y') \quad (3)$$

Where a>1, m, n represent the scale and orientation, and x',y' are calculated by :

$$x' = a^{-m}(x\cos\theta + y\sin\theta) \quad (4)$$

$$y' = a^{-m}(-x\sin\theta + y\cos\theta) \quad (5)$$

Where $\theta = \frac{n\pi}{N}$ and N denotes the total number of orientations.

For the reference image R(x,y), its Gabor wavelet transform is computed as:

$$W_{m,n}(u,v) = \int_\Omega R(x,y)g^*_{m,n}(u-x, v-y)dxdy \quad (6)$$

where $g_{m,n}$ is the self-similar filter dictionary, and $g^*_{m,n}$ denotes the complex conjugate. Then we compute the mean $\mu_{m,n}$ and the standard deviation $\sigma_{m,n}$ as the coefficients:

$$\mu_{m,n} = \int\int \mid W_{m,n}(x,y) \mid dxdy \quad (7)$$

$$\sigma_{m,n} = \sqrt{\int\int (\mid W_{m,n}(x,y) \mid - \mu_{m,n})^2 dxdy} \quad (8)$$

We use the $\mu_{m,n}$ and $\sigma_{m,n}$ constitute the feature vector. In our experiment we set scales m=3 and orientations n=8, so our feature vector is $[\mu_{0,0}\ \sigma_{0,0}\ \mu_{0,1}\ \sigma_{0,1}\ \ldots\ \mu_{2,7}\ \sigma_{2,7}]^T$.

After we construct feature vector for each pixel in the reference image, which forms the feature space, given the pixel in the input image, we can calculate its feature vector, find the nearest point in the reference image and assign its class number to the grayscale point. This method is simple, but result is not good. So we use K-nearest-neighbor (Knn) method. Knn is a method for classifying pixels based on closest training examples in the feature space. It is a simple learning technique: a pixel is classified by a majority vote of its neighbors, with the pixel being assigned to the class most common amongst its k nearest neighbors. However, as the feature space is a high dimensional data, it may import erroneous result. So we use LDA method to discriminate the different classes. To refine the classifier we subsequently apply a PCA projection, which maximizes the difference between the different classes and minimizes the difference within the same class. Finally, we use Knn method to assign a class number for every pixel in the input image.

Figures 4(a-b) display the classification results of Irony's method and our method. We can see from the figures that Iorny's result has great error, most pixels of the wolf's hat have been misclassified to the background class. While, our method avoids this phenomenon, and produce a much better and more correct result.

(a) classification result using Irony's method (b) classification result of our method (c) classification result after refinement

Figure 4. The comparison of classification results

4.2 Colorization

Although we can obtain a satisfactory result after the steps above, there are still some errors (shown in Fig. 4(b)). We can improve our result considering the spatial consistency. For every pixel in the input image, its neighbor may contain different class numbers; we compute its reliability of all the possible class numbers, take the maximum reliability value and corresponding class number. Then we replace the class number for each pixel to generate the final result of classification. The reliability of pixel p is defined as:

$$reliability(p,c) = \frac{\sum_{q\in N(p,c)} W_q}{\sum_{r\in N(p)} W_r} \quad (9)$$

where reliability(p,c) represents the reliability value of class number c. N(p,c) is the neighbor pixels of p whose class number is c. Weight w_q depends on the distance $D(q, R_q)$, R_q is the best match in the reference image, having the same class number with p. The weight is defined as:

$$W_q = \frac{\exp(-D(q, R_q))}{\sum_{r \in N(q)} \exp(-D(r, R_r))} \tag{10}$$

The reliability is high for the pixels where all (or most) its neighbor are in the same class, and low for the pixels who are around the boundaries.

The refined result of classification is shown in Fig 4(c). Compared to Fig. 4(b), many erroneous pixels have been corrected such as the pixels inside the wolf's hat.

After refining the classification result, we are going to get the color from the reference image. We work in YUV color space, where Y is the monochromatic luminance channel, while U and V are the chrominance channels. We use Col (p) denotes the color value of pixel p, which is defined as:

$$Col(p) = \sum_{q \in N(p,c)} W_q Col(R_{q \to p}) \tag{11}$$

where $Col(R_{q \to p})$ stands for the pixel in the reference image whose position relative to R_q is the same as the position of p relative to q. The color of pixel p is a weighted average of the neighbor pixels that are in the same class.

So far we have got a colorized image, but some pixels are colorized mistakenly, in order to improve the colorization result, we leave the more reliable pixels and use these pixels as constraint applying the optimization-based colorization algorithm to optimize the final result. To leave the reliable pixels, we set a threshold, and only leave the pixels whose reliability value are greater than the threshold. In our experiment, the threshold is fixed to 0.75. Depending on the pixels which have the high enough reliability value as scribbles, we then use the method of Levin et al. solving a least-squares optimization problem to obtain the final result. The final result is shown in Fig. 5.

Figure 5. The final colorization result of our method

5. Experimental Results

All the examples shown in our paper are derived using three scales and eight orientations for texture analysis, so our initial feature space has 48 dimensions. After applying LDA and PCA projection, we reduce the dimensions of the feature space to 10, preserving the most discriminating feature vectors. We set the parameter k to 7 of the Knn method, calculating the k nearest neighbors and choosing the majority class number. For the scribble generation, we reserve the pixels whose reliability are greater than 0.75. All the examples are computed on a computer equipped with an Intel(R) Core(TM) i5 CPU 760 @2.80 GHz and 4.00 GB system memory.

In Fig. 5, we shows the result of our method applying to a manga image. Additionally, our method is adequate to the natural image. Figure 6 exhibits the result of colorizing a horse using our method. Compared with the method of Levin and Qu, our method is more convenient, requiring less manual work. But the advantage of their methods is they do not need a reference image, in order to improve this, we can colorize a part of the image using scribbles as reference image and then apply our method. This solution is particularly effective to the image that contains lots of textures. On the other hand, adopting the Gabor wavelet analysis, our method consider the pattern consistency, and generate better classification result than Irony's(see the Fig. 4(a-b)).

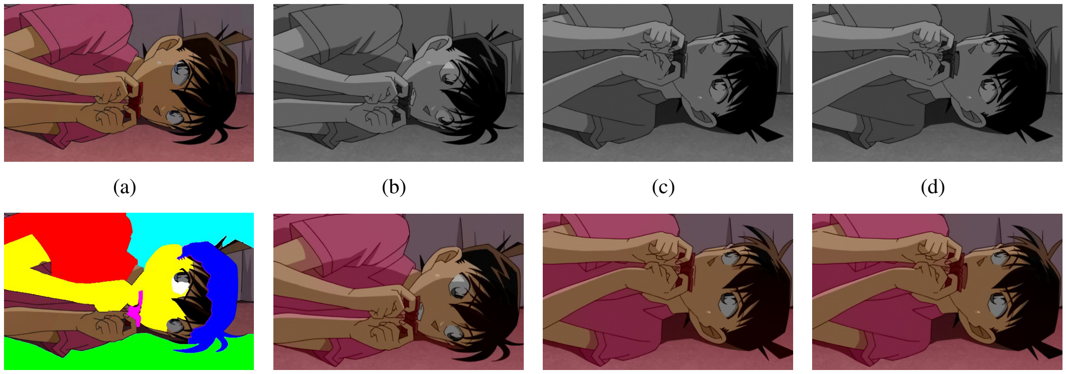
(a) reference image (b) grayscale image (c) colorization result

Figure 6. Apply our method to natural image

Figure 7 shows the colorization result of videos. Our method is advantageous to colorize video, since we can colorize a typical frame, and all the rest frames will be colorized automatically. Figures 7(a,e) is the reference chromatic frame and its segmentation. Figures 7(b-d,f-h) is three frames intercepted from the movie and their colorization results.

6. Conclusion

In this paper, we present a novel method to colorize a grayscale image from the reference image. We use



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7. Apply our method to video clip extracted from Detective Conan

the Gabor wavelet to analyze the texture, considering the pattern continuity. We train a classifier to classify the pixels in the grayscale image into different classes that has been pre-segmented by the user in the reference image. To enhance the spatial consistency, we implement a step to refine the wrong pixel class number by the class number having the most reliability. We leave the most likely correctly colorized pixels to produce the scribbles and use the optimization-based colorization algorithm to gain the final result. Our method need little human interactive, in the meanwhile, generate a much better colorization result.

Our method is applicable to all kinds of images, especially for the images full of textures, such as cartoons. Moreover, our method is more beneficial for the video colorization, since once a frame of the video is colorized, all the rest frames can be colorized automatically without human intervention. In conclusion, no matter for natural images or cartoons, images or videos, our method can provide a high-quality colorization result automatically.

Acknowledgment

This work was partly supported by funds from the National Basic Research Project of China (No. 2011CB302200), the National Natural Science Foundation of China (No.60873136) and the Innovation Program of the Science and Technology Commission of Shanghai. Municipality (No. 10511501200).

References

[1] W. Markle and B. Hunt, “Coloring a black and white signal using motion detection,” Jul. 5 1988, uS Patent 4,755,870.

[2] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 689–694.

[3] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 277–280.

[4] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *Computer Graphics and Applications, IEEE*, vol. 21, no. 5, pp. 34–41, 2001.

[5] R. Irony, D. Cohen-Or, and D. Lischinski, “Colorization by example,” in *Proceedings of the Eurographics Symposium on Rendering*, 2005, pp. 201–210.

[6] D. Sỳkora, J. Buriánek, and J. Žára, “Unsupervised colorization of black-and-white cartoons,” in *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*. ACM, 2004, pp. 121–127.

[7] L. Yatziv and G. Sapiro, “Fast image and video colorization using chrominance blending,” *Image Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1120–1129, 2006.

[8] Y. Qu, T. Wong, and P. Heng, “Manga colorization,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 1214–1220, 2006.

[9] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y. Xu, and H. Shum, “Natural image colorization,” in *Rendering Techniques 2007 (Proceedings Eurographics Symposium on Rendering)*. Citeseer, 2007.