Pattern Recognition Letters 33 (2012) 1673–1681

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Automatic grayscale image colorization using histogram regression

Shiguang Liu *, Xiang Zhang

School of Computer Science and Technology, Tianjin University, Tianjin 300072, PR China

ARTICLE INFO

Article history:
Received 6 November 2011
Available online 15 June 2012

Communicated by S. Sarkar

Keywords:
Image editing
Colorization
Histogram regression
Colorization confidence

ABSTRACT

Colorization aims to adding colors to a grayscale image. This task is ill-posed in the sense that assigning the colors to a grayscale image without any prior knowledge is ambiguous. Most of the previous methods require some amount of user interventions, making colorization a hard work. Motivated by this, a novel automatic grayscale image colorization method based on histogram regression is presented in this paper. A source image is adopted to provide the color information. Locally weighted regression is performed on both the grayscale image and the source image. Thus, the feature distributions of two images can be obtained. Then, a new matching method is proposed to align these features by finding and adjusting the zero-points of the histogram. When the luminance-color correspondence was achieved, the grayscale image is colorized in a weighted way. Moreover, a new evaluation method is specially designed to assess the confidence of the colorization results. Various experiment results are given to show the validity of this method.

1. Introduction

Colorization is a computer-assisted process of giving colors to a grayscale image or video, aiming at increasing the visual appeal of images such as old black and white photos, classic movies or scientific illustrations, etc.

The task of colorization involves assigning chromatic values to a gray-scale image. Since different colors may have the same intensity, colorization has no exact solution. Most of the current approaches (Levin et al., 2004; Irony et al., 2005; Luan et al., 2007) require an amount of user interactions to achieve a satisfactory result, making colorization a tedious handmade work. Also, it is difficult to evaluate the colorization results using previous methods. Among them, color transferring methods (Welsh et al., 2002; Reinhard et al., 2001; Abadpour and Kasaei, 2007) and image analogy methods (Hertzmann et al., 2001) are widely used. These methods need the user to provide a reference color image, which is called the source image. The grayscale image to be colorized is called the target image. These kinds of methods can achieve more plausible results if the contents of the source image and the target image are similar. However, they are labor intensive and time consuming as the user must match the source image and the target image manually. To solve this problem, this paper proposed a novel automatic method to realize colorization of grayscale images.

To realize automatic colorization of grayscale images, we introduce machine learning based methods. A source image with the similar contents with the grayscale image is provided to achieve plausible results. The new colorization method includes the following steps. First, locally weighted regression (LWR) on the histograms of grayscale image and the source image is performed to analyze their feature distributions. Then, we propose a novel method to match these features by finding and adjusting the zero-points. Here, the zero-points are the local extreme of the histogram fitting curve generating using LWR. After the matching processing, we transfer the colors from the source image to the grayscale image automatically. To evaluate the quality of the colorization results, the conception of colorization confidence is introduced at last. The main contributions of our method can be summarized as follows:

- We propose the method of locally weighted regression on the histogram for analyzing and clustering the image. This method is automatic and efficient.
- A novel method for achieving luminance-color correspondence by finding and adjusting the zero-points is specially designed.
- We present the conception of colorization confidence to evaluate the colorization results quantitatively.
- Our method is easy to implement, having the potential to deal with images or videos massively.

The rest of this paper is organized as follows. In Section 2, we introduce the framework of our algorithm. Section 3 gives a brief survey of related work. Then we give the method of locally weighted regression on the images in Section 4. In Section 5, we propose the method for zero-points adjustment and matching. Section 6 discusses the weighted colorization process of grayscale images. Section 7 describes the method of colorization evaluation. The implementations and results are discussed in Section 8. Conclusions and future work conclude the paper.

* Corresponding author.
E-mail addresses: shgliu@126.com, lsg@tju.edu.cn (S. Liu).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.06.001

2. Related work

Colorization has been extensively studied in recent years. Various methods have been proposed to solve this challenging task. However, most of them are semi-automatic methods which require some amounts of user interactions.

Gonzalez and Woods (1987) proposed the method of luminance keying to transfer color to a grayscale image. This method utilizes a user-defined look-up table to assign a color to each grayscale value. When applying different colors at the same intensity level, the user should simultaneously use a few luminance keys for different regions manually, making the process very tedious. Welsh et al. (2002) introduced a colorization method by transferring color from a source color image to a target grayscale image via matching color information between the images. It is inspired by the method of color transfer between images (Reinhard et al., 2001) and the idea of image analogy (Hertzmann et al., 2001). This method needs the user to match the areas using swatches. Levin et al. (2004) introduced an interactive colorization method based on the premise that nearby pixels in space that had similar gray levels should also have similar colors. Without considering the boundary and region information, unseemly color sometimes leaks from one region to others. Nie et al. (2007) improved (Levin et al., 2004) and presented an optimization based interactive grayscale image colorization method. It is reported that this method gives the same good quality of colorized images as the method of (Levin et al., 2004) with a fraction of the computational cost. Nie et al. (2005) proposed a grayscale image colorization method based on a local correlation based optimization algorithm. However, this method is restricted to some assumptions about the color correlativity between pixels in different regions. Irony et al. (2005) presented a method for colorizing grayscale image by transferring color from a segmented example image. Rather than relying on a series of independent pixel-level decisions, they accounted for the higher-level context of each pixel and thus achieved a higher degree of spatial consistency. Yatziv and Sapiro (2006) proposed a method for image and video colorization using chrominance blending. This method is computationally simple and effective. However, to realize desired results, it needs the user to mark some chrominance scribbles. Taking advantages of the principal component analysis, Abadpour and Kasaei (2007) introduced a method for grayscale image colorization. It can also be used to recolor colored images. However, this method suffers from complicate segmentation which is performed manually using the *magic select* tool in *Adobe Photoshop*. Luan et al. (2007) presented an interactive system for colorizing the natural images. The colorization procedure is separated into two stages: color labeling and color mapping. This method is effective for natural image colorization. However, to obtain good colorization results, the user had to draw multiple stokes on similar patterns with different orientation and scales. Liu et al. (2008) proposed an example-based colorization technique considering illumination differences between grayscale image and color reference images. This method need to search suitable reference images from the web. Xiang et al. (2009) proposed a selective color transfer method. With this method, a target image can be colorized by referring to more than one source image. Motivated by the setting of reproducing kernel Hilbert space (RKHS), Quang et al. (2010) proposed a RKHS framework for image and video colorization. This method can achieve realistic results. However, it needs initializing the colors for different areas manually by the user. It is labor-intensive if there are a large amount of different contents in an image. Some researchers also extended the colorization method to cartoon images and manga. Sýkora et al. (2003) presented a method for colorization of black and white cartoons. They combined the methods of image segmentation, patch-based sampling and probabilistic reasoning to achieve the colorization results. Lagodzinski and Smolka (2008) presented a novel colorization method which takes advantage of the modified morphological distance transform to propagate the color scribbled by a user on the grayscale image. Experiments showed that this method can generate good results, with potential to be applied for the virtual restoration of works of art. Kawulok and Smolka (2010) proposed a new method for image colorization based on manually added scribbles, using the competitive approach for selecting an appropriate type of the path cost. This method can yield better results compared with other colorization techniques especially for highly-textured images. Qu et al. (2006) studied the colorization method of manga. Though the above methods can realize colorization of different grayscale images, they all require the user's interactions to provide additional information. These interactions restrict them to deal with images/videos massively. Until now, few automatic grayscale image colorization methods have been reported in the field of computer graphics.

Charpiat et al. (2008) proposed an automatic image colorization method via multimodal predictions. However, to achieve good results, it may be corrected by user-provided color landmarks. Li and Hao (2008) proposed a learning-based method for colorization of grayscale images. This method extracts the input (grayscale) and the output (chromatic) samples as feature vectors from training data, views them as distributed in two manifolds with similar structure. It is automatic which can be used for batch processing. Morimoto et al. (2009) realized automatic colorization of grayscale images using multiple images on the web. This method depends on the searching results from millions images collected from the web. It will produce unnatural results due to the source images that are structurally similar but semantically different. To overcome these shortcomings, we proposed a novel automatic grayscale image colorization method using a source image provided by the user.

Another topic related to our method is histogram matching. Heeger and Bergen (1995) used histogram matching to achieve pyramid-based texture analysis/synthesis. Cheng and Sun (2000) extended the general idea of a histogram to homogeneity domain and adopted it to implement color image segmentation. Neumann and Neumann (2005) realized color transfer using 3D histogram matching in the HSL color space. This method can achieve an exact match of the gamut between the source image and the target image. Similarly, Xiao and Ma (2009) adopted histogram matching in the $l\alpha\beta$ color space to preserve the gradients of the images when performing color transferring. To address the problem of eliminating unwanted color variation between similar image pairs, Senanayake and Alexander (2007) proposed a color transfer method by feature based histogram registration. The source and target histograms are aligned based on corresponding features that are persistent peaks through scale space using a polynominal mapping. However, this method will fail when the histograms of the image pairs are widely different. Recently, Pouli and Reinhard (2010) improved this method by expanding on the notion of histograms in a scale space, which is capable of matching the features of the source image to a selected level of accuracy to yield a variety of creative color transfer effects. To reduce the expense of computing histogram properties over large neighborhoods, Kass and Solomon (2010) presented an efficient and practical method for computing accurate derivatives ad integrals of locally-weighted histograms.

3. Overview of the histogram regression based automatic colorization method

In the novel algorithm, we first perform locally weighted linear regression on the luminance histograms of both source and target

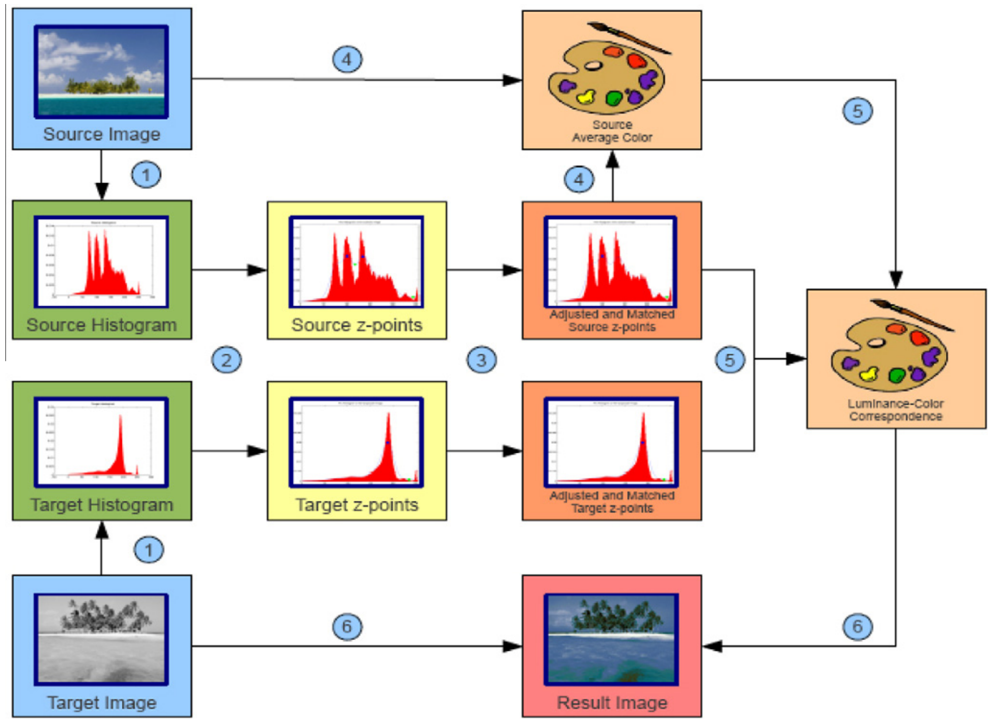

Fig. 1. Overview of our algorithm.

image. With the slopes in the regression result, we can detect zero-points (i.e., local maxima and minima) of the approximated histogram, and then adjust these points so that they can be matched between source and target. Then, with the calculated average color from the source image, we can get the luminance-color correspondence for the target image in a weighted way, and the weights are determined by the adjusted zero-points. The colorization result is achieved by mapping this luminance-color correspondence with the target image directly. Fig. 1 gives the framework of our algorithm.

4. Locally weighted regression on histograms

The histogram of an image represents the number of pixels in different luminance intervals. If we could find matched points in the two histograms between the grayscale image and the source image, it is possible to get a solution to correspond the colors of the grayscale image to the colors of the source image. To realize the match in a continuous way, it is needed to characterize the histogram with a curve. Here, we apply locally weighted regression to the histograms of the source image and the grayscale image to generate the fitting curves (see Fig. 2). Below, we will discuss this method.

Locally weighted regression performs a regression around a point of interest using only training data that are local to that point. It is a non-parametric approach which explicitly retains the training data, and it can be used each time a prediction needs to be made. This is a minimization problem which can be expressed as follows:

$$\theta = \arg\min \sum_{j=1}^{m} w_j(\theta^T X_j - Y_j), \tag{1}$$

where

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}, \quad X_j = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \tag{2}$$

Here, (X, Y) and θ represent the training data and the fitting data, respectively, (X_j, Y_j) denote the jth sampling of the training data, m is the size of the training data. The vector θ is the regression results for a particular j, θ_0 is the y-intersect at j, and θ_1 is the slope at j. w_j is a non-negative weight. Intuitively, if w_j is large for a particular value of j, we'll try to make $(\theta^T X_j - Y_j)$ small when picking θ. On the other hand, if w_j is small, then $(\theta^T X_j - Y_j)$ will be pretty much ignored in the fitting process. We calculate this weight term as follows:

$$w_j = \exp\left(-\frac{\|X_j - X\|^2}{2\tau^2}\right), \tag{3}$$

where τ is a constant in our experiments. Its value is 10.In our special colorization case, we denote the histogram of an image as Y_j $(j \in N,\ 0 \leqslant j \leqslant 255)$ and

$$X_j = \begin{pmatrix} 1 \\ j \end{pmatrix}, \tag{4}$$

Supposing that the luminance is represented in an interval $[0, 255]$, Y_j indicates the number of pixels whose luminance is j. As shown in formula (1), locally weighted regression is to perform local regression for every data X with a weighted consideration of other data's effects on the fitting procedure of it. By defining other two matrices as follows, the result of locally weighted regression can be derived as $\theta = (\bar{X}^T\bar{X})^{-1}\bar{X}^T\bar{Y}$

$$\bar{X} = \begin{bmatrix} (X'_1)^T \\ (X'_2)^T \\ \cdots \\ (X'_m)^T \end{bmatrix}, \bar{Y} = \begin{bmatrix} Y'_1 \\ Y'_2 \\ \cdots \\ Y'_m \end{bmatrix}$$

Here, $X'_j = \sqrt{w_j}X_j$ and $Y'_j = \sqrt{w_j}Y_j$. Fig. 2 shows the results of locally weighted regression. Fig. 2(a) is the grayscale image. Fig. 2(b) gives the histogram of the grayscale image and the fitting curve which was drawn in blue. Fig. 2(c) and (d) shows the values of θ_0 and θ_1, which represent the slopes and the y-intercepts of the fitting curve, respectively. By this fitting curve, we can calculate the

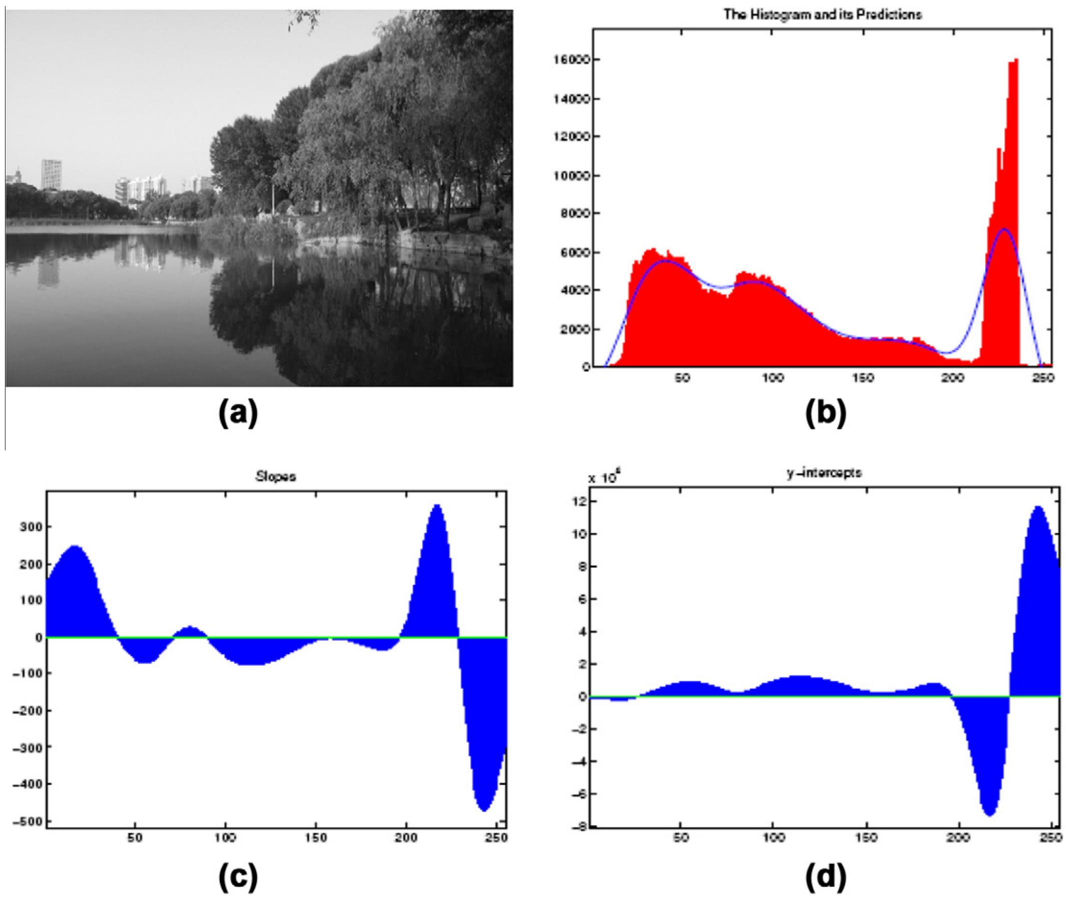

Fig. 2. The results of performing LWR on the image histogram: (a) the grayscale image, (b) the histogram and the fitting curve, (c) the slopes and (d) the y-intercepts.

local extreme points of the histogram which will be used for segmentations. In our method, the local extreme points are obtained by calculating the points which have zero-slopes as the slopes behave just like the first derivatives of the histogram function.

5. Zero-points adjustment and matching

With the assumption that the contents of the source image and the grayscale image are similar, we can further interpret this similarity in terms of the similarity of their histograms. In fact, the similarity of the histograms can be expressed by its local extreme (zero-points), more specifically, the number of the extreme and its minima–maxima alternation. We adjust the number of the extreme and its minima–maxima alternation of two histograms to be the same, thus we can match the corresponding colors of both images.

5.1. Generation of zero-points

The zero-points are generated by iterating through all the luminance values, determining if any of a point is a local minimum or local maximum. If a point is a local extremum, its first derivative should be 0. For the discrete form of regression results, we achieve by finding the slopes that are nearest to 0 to determine possible local extrema. We determine the point pairs that have arbitrary comparison relationships with 0, and find the nearest local extrema by choosing the nearer-to-0 point in a candidate pair. If the candidate pair of local extrema is ascending, then it is a local minimum; otherwise, it is a local maximum.

From the above procedure, we obtain two sequences. One is the zero-points sequence Z, and the other is the extrema alternation sequence T. The values in T indicate the alternation of maxima and minima of Z. If $T(i) = 1$, then $Z(i)$ is a local maximum; if $T(i) = -1$, then $Z(i)$ is a local minimum. The histogram segmentation can be directly carried out by segmenting the histogram with these local extreme points (zero-points). We denote this segmentation set as S. We later use these segmentations for colorization. This procedure for generation of zero-points is called the zero-points generation algorithm.

When applying the above zero-points generation algorithm to the histograms of both the grayscale image and the source image, we can get the histogram segmentations for them. We denote Z_S and T_S as the result sequences for source image, and Z_T and T_T for the grayscale image, respectively. Similarly, we denote S_S and S_T for the histogram segmentation set for the source image and the grayscale image.

If $T_S = T_T$, the contents of the source image and the grayscale image are the similar. This also interprets that the similarity of the histograms represents some similarity of the images. Under this condition, the segmentations S_S and S_T are matched naturally and orderly. On the contrary, if $T_S \neq T_T$, we should adjust the sequence sets T and Z to make them matched for colorization. Our basic idea is to eliminate points in the set with larger number of zero-points. Also, by recording the eliminated points, we can further represent the similarity of the two images qualitatively. Next, we will discuss how to adjust the zero-points sets.

5.2. Adjustment of Zero-points

From the above part, we know that the element in sequence T is an alternation of 1 and -1. Starting from the first element in sequence T, 1 and -1 appears in turn. So, in order to adjust the sequence T, we need to preserve this alternation property. We will first adjust the front and rear part of the sets. This is necessary if the difference number of the sizes of the sequences is not even. The following algorithm does this job. Assume A is a sequence or a set, $\delta(A)$ indicates the size of A.

The purpose of front-rear adjustment is to make the difference of the size of the two zero-point sequences even, so that we can later delimit pairs of points from the larger sequence (i.e., pair adjustment) without changing its alternation property. The scheme of the whole algorithm is shown as the above. Specifically, it can be summarized as follows. First, the front-rear adjustment algorithm will test if the difference of the sizes of the two sequences is odd. If it is odd, elimination of a front or rear point is needed. Then, the algorithm selects the larger sequence. In addition, the algorithm compares the front and the rear of the sequences. If any of them is not equal, the algorithm eliminates it in the larger sequence. The result of this algorithm will make the difference of the two sequences even. As a consequence, we can later eliminate two points at once from the larger sequence without changing the alternation property.

After the front-rear adjustment, we will match the zero-points in the front and rear part of the sequence. Though the sequences are still not matched with each other, the difference between the sizes of them is even. We can eliminate the zero-points evenly from the larger set in order to match the sequences. We propose a novel iterative algorithm called pair adjustment algorithm to realize this, which eliminates two adjacent points in each iteration.The elimination of points should not be performed randomly. The way to choose which adjacent pair of points is to be eliminated in each iteration, is decided by its effect on the changing of the sequences. We eliminate the pair of points that change the trend which the zero-points sequences Z has less number of elements. Fig. 3 shows pairs of points inside an ascent or descent trend in Z. Fig. 3(a) and (b) shows the zero-points in an ascent and descent order in one trend, respectively. The points in blue and green represent the local maximal and minimal points in the sequences, respectively. As Fig. 3 shows, both in the cases of 3(a) and (b), the inner pairs of points are the candidates for elimination. If we eliminate them, the overall trend of the sequence does not change greatly.

We find these points by comparing each pair of points with its adjacent points. Assume we have z_i, $z_{i+1} \in Z$, and $g(z_i) > g(z_{i+1})$. If $g(z_i) < g(z_{i+2})$ and $g(z_{i-1}) < g(z_{i+1})$, then the point pair (z_i, z_{i+1}) is in an ascent trend. Similarly, we can obtain the same method to find point pairs that are in a descent trend. After finding all pairs of such points, we compare their differences. The pair of points with the smallest difference will be eliminated. That is to say, for a pair of adjacent points. $z_i, z_{i+1} \in Z$, if $|g(z_i) - g(z_{i+1})|$ is the smallest, this pair of points will be eliminated in this iteration of the algorithm. There is also another case that no pair of points exists in a trend. In such cases, we only eliminate the pairs of points that have the smallest difference.

In the pair adjustment procedure, we will first append the minimal and maximal luminance values to the front and rear part of the larger sequence, so that the procedure afterwards will not exceed the array boundary. Note that we do not consider these two points as candidates for deletion. Then, the algorithm iterates till the difference between the size of the sequences becomes 0. In each iteration, we will make sure there will be a pair of points deleted. In any of the iteration, if there exist pairs of points that are in a trend, we will first delete the pair of such points with smallest difference in histogram values to match the zero-points in an optimal way. If no in-trend pairs exist, then we just delete the pair of points with smallest difference in histogram values. The way we determine if a pair of points is in a trend, is to perform comparison of them with an extra two points that are adjacent to them.

In the previous steps, we have made the difference of the sizes of the sequence sets even. Thus, we can eliminate two points in each iteration. Front-rear adjustment algorithm and pair adjustment algorithm together are called the zero-points adjustment algorithm. By this algorithm, we can adjust the two histograms to be matched.

6. Weighted colorization

After we get the matched histograms of the source image and the grayscale image, we will colorize the grayscale image by borrowing color from the corresponding parts in the source image. We define c as a color, $c(u, v)$ as the color of an image at the coordinate (u, v), and $U \times V$ as the size of the image. The luminance of a color c is represented as a function $L(c)$.

Our colorization algorithm consists of two steps. First, we calculate the average color $C(j)$ for each luminance j of the source image. Then, we perform colorization for each matched histogram segmentation pair. The fist step can be done using an average operation. For convenience we denote $C_S(j)$ as the computed average color for luminance j from the source image. For the second step, we will borrow the weight function in locally weighted regression to construct a special weighted algorithm. For each point with

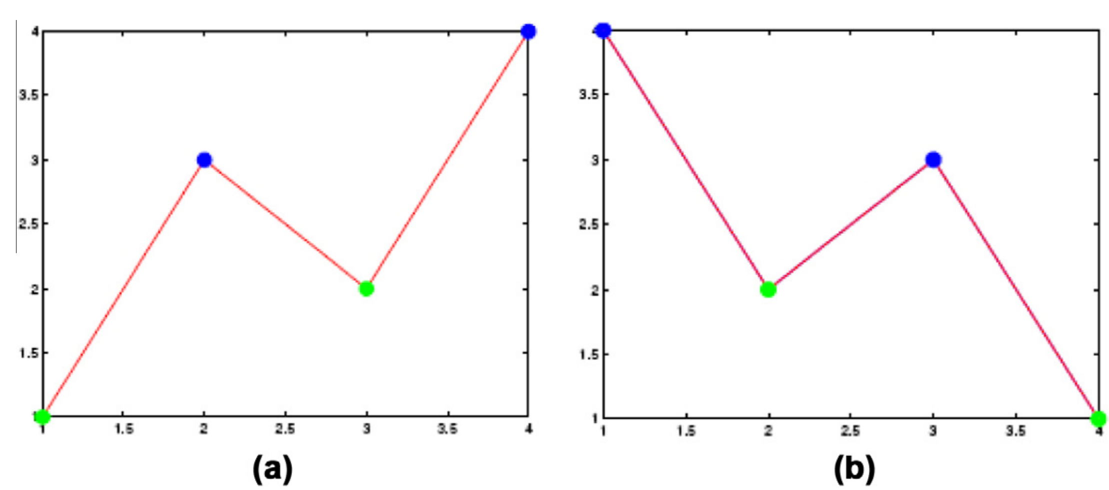

Fig. 3. Pairs of points in a trend: (a) zero-points in ascent trend and (b) zero-points in descent trend.

luminance j_T on the grayscale image, we will colorize it using the weighted average of corresponding average color $C_S(j_S)$. The weight function is shown as follows:

$$w(x,\mu,\tau) = \exp\left(-\frac{\|x-\mu\|^2}{2\tau^2}\right),$$
$$\mu = (j-\eta_T)\frac{\delta(S_S)}{\delta(S_T)} + \eta_S, \quad \tau = \frac{\delta(S_S)}{\delta(S_T)}, \tag{5}$$

where μ and τ are the central value and the bandwidth, respectively. η_S and η_T are the minimal value in S_S and S_T, which represent the luminance segmentation we have achieved using regression on histograms. From the formula above we can actually recognize that this weighted average color is a Gaussian-like smoothing procedure for the luminance-color correspondence, but it will not change the texture of the image. The weighted luminance-color correspondence is only a weighted average for the colors, which ensures that two adjacent luminance values should not have abrupt color difference in the result. Also, it is possible to preserve the luminance of the original image (and this will also ensure that the texture is not affected) using a luminance-separated color space, such as YCbCr, which compute and colorize using only the color channels, where in the case they keep Cb and Cr but not Y. In a summary, for two matched segmentation S_S and S_T, the weighted colorization formula can be expressed as:

$$C_T(j) = \frac{\sum_{i=1}^{255} w\left(i, (j-\eta_T)\frac{\delta(S_S)}{\delta(S_T)} + \eta_S, \frac{\delta(S_S)}{\delta(S_T)}\right)\cdot C_S(i)}{\sum_{i=1}^{255} w\left(i, (j-\eta_T)\frac{\delta(S_S)}{\delta(S_T)} + \eta_S, \frac{\delta(S_S)}{\delta(S_T)}\right)}, \tag{6}$$

where $C_T(j)$ is the average color of the grayscale image. In this algorithm, the central value and the bandwidth are μ and τ, respectively. These calculations can make the transformation of the values from S_T to that of S_S linearly. The bandwidth also ensures the smoothness of the histograms easily and efficiently. For each pixel (u,v) with luminance j in the grayscale image, we set $c_T(u,v) = C_T(j)$.

7. Confidence

By zero-points adjustment algorithm, we can get several useful data. One of them is the difference of sizes of sequence sets δ. The larger is δ, the more dissimilar are the two images. So our first way to get the confidence measurement is to measure the size of δ, which is expressed as the following formula:

$$q_1 = e^{-\frac{\delta}{\delta(Z_T)}}, \tag{7}$$

Note that in the above equation, Z_T is the sequence set of the grayscale image after the processing by zero-points adjustment algorithm. So $\delta(Z_T)$ is equal to $\delta(Z_S)$. Z_T and Z_S (both are obtained using the algorithm of zero-points adjustment) can also be used to measure the similarity of the two images. We define λ and σ as follows:

$$\lambda = \frac{1}{256}\sum_{j=0}^{255} y_j, \tag{8}$$

$$\sigma = \sqrt{\frac{1}{256}\sum_{j=0}^{255}(y_j-\lambda)^2}, \tag{9}$$

where λ and σ are the mean and standard deviation of the histogram values. We define the normalized sequence of histogram values G as $G(j) = (g(Z(j)) - \lambda)/\sigma$. Let G_S and G_T be the normalized sequence of histogram values of the source image and the grayscale image, which can be regarded as two vectors. We represent the similarity of two normalized vectors as the cosine of their angles. We calculate another representation q_2 of the similarity as $q_2 = G'_S G_T / (\|G_S\| \cdot \|G_T\|)$, here G'_S is the transpose of G_S, $\|\cdot\|$ represents the norm of a vector. We then combine q_1 and q_2 together to express the final measurement for the similarity of two images: $q = q_1 q_2$. We call this similarity value the colorization confidence. The larger is it, the more confident is the colorization result. This interprets the intuition that the more similar the two images are, the better the algorithm should colorize the image. In practice, this confidence value is achieved by considering the similarity of histograms between the two images, by measuring their zero-points difference.

8. Implementation and results

We have made some experiments based on the above algorithm. All the experiments are run on a common PC with Pentium Dual Core 2.0 GHz, 1G RAM and the operating system is Microsoft Windows XP Professional Service Pack 2. The programs are written in Matlab. Table 1 collects some parameters with our method.

Fig. 4 shows the results of our algorithm. Fig. 4a–e represents the source image, the grayscale image, the matching result of the source image, the matching result of the target image and the colorization result, respectively. Fig. 4(f) and (g) is the histograms of the source image and the grayscale image. Using our zero-points adjustment algorithm, we obtained the adjusted zero-points sequence of the histograms of the source image and the grayscale image shown as Fig. 4(h) and (i), respectively. Fig. 4(j) is the histogram of the colorization result. From Fig. 4, we can see that there are four and three local extreme points in the histogram of the source image and the grayscale image, respectively. By processing them using the zero-points adjustment algorithm, we obtained the same number of the segmentations for colorization shown in Fig. 4(h) and (i). As shown in Table 1, the colorization confidence of this experiment is 99.81%. Fig. 4 shows another colorization result of a lake scene. From Fig. 4(e), we can see that the colorization result is satisfactory. The leaf areas are colorized correctly according to the source image. Our method can realize the glossy and reflective effects of the water surface are colorized plausibly.

Some colorization methods (Levin et al., 2004; Irony et al., 2005; Luan et al., 2007) need the user to interactively scribble on the images. Though these kinds of methods can achieve good results, the interactions with the image requires professions, thus restrict

Table 1
Parameters and their values in our experiments.

Images	Resolution of the source image	Resolution of target image	Time (s)	Colorization confidence
Fig. 4	900 × 600	900 × 600	39.95	0.99
Fig. 6(a)	560 × 420	560 × 420	34.68	0.96
Fig. 6(b)	800 × 462	800 × 462	36.43	0.95
Fig. 6(c)	666 × 500	666 × 500	36.95	0.93
Fig. 6(d)	500 × 420	500 × 420	12.27	0.94
Fig. 6(e)	986 × 598	986 × 598	35.16	0.99
Fig. 6(f)	666 × 500	666 × 500	22.74	0.92

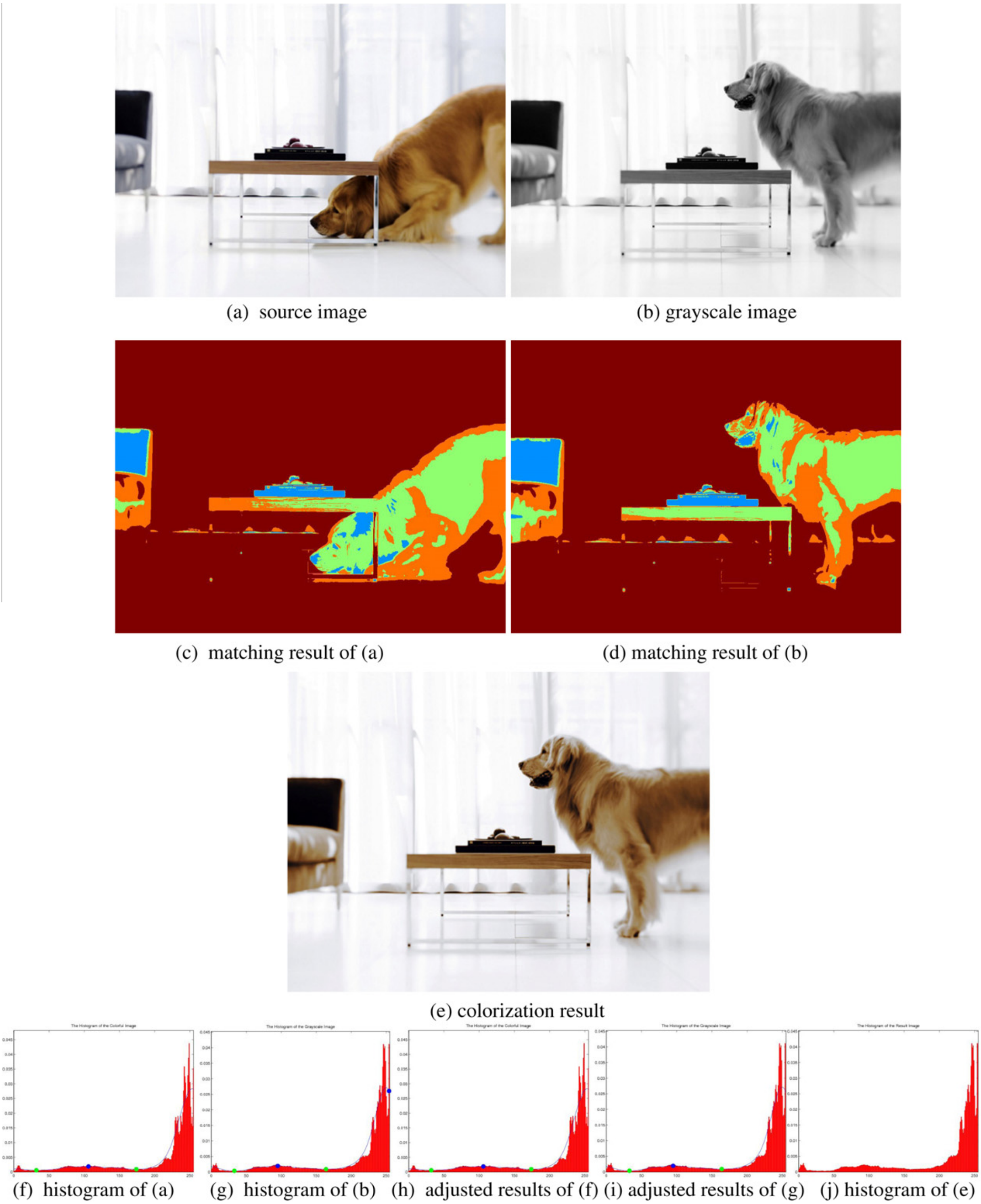

Fig. 4. Colorization result of an animal scene.

to common users. In contrast to them, our new method can realize the colorization automatically, without any scribbling or other interactions. Fig. 5 shows the contrast of the colorization result of our method and other methods. The images in the first two columns of Fig. 5(a) and (b) are the source images and the grayscale images, respectively. The images in the last columns are the ground truth. The images in the third columns through the fifth column are the colorization results of Welsh's method (Welsh et al., 2002), Li et al.'s method (Li and Hao, 2008) and our method. Fig. 5(c) gives the PSNR (Peak Signal to Noise Ratio) contrast among the above methods. In contrast to the above methods, our algorithm can obtain the similar image intensity and contrast to the ground truth. More colorization results of various scenes are shown in Fig. 6.

Limitations. From the sample results, we see that our algorithm can realize good results for most images. However, there are still limitations for this method. The intuitive reason is that only considering luminance distribution without regarding the spatial

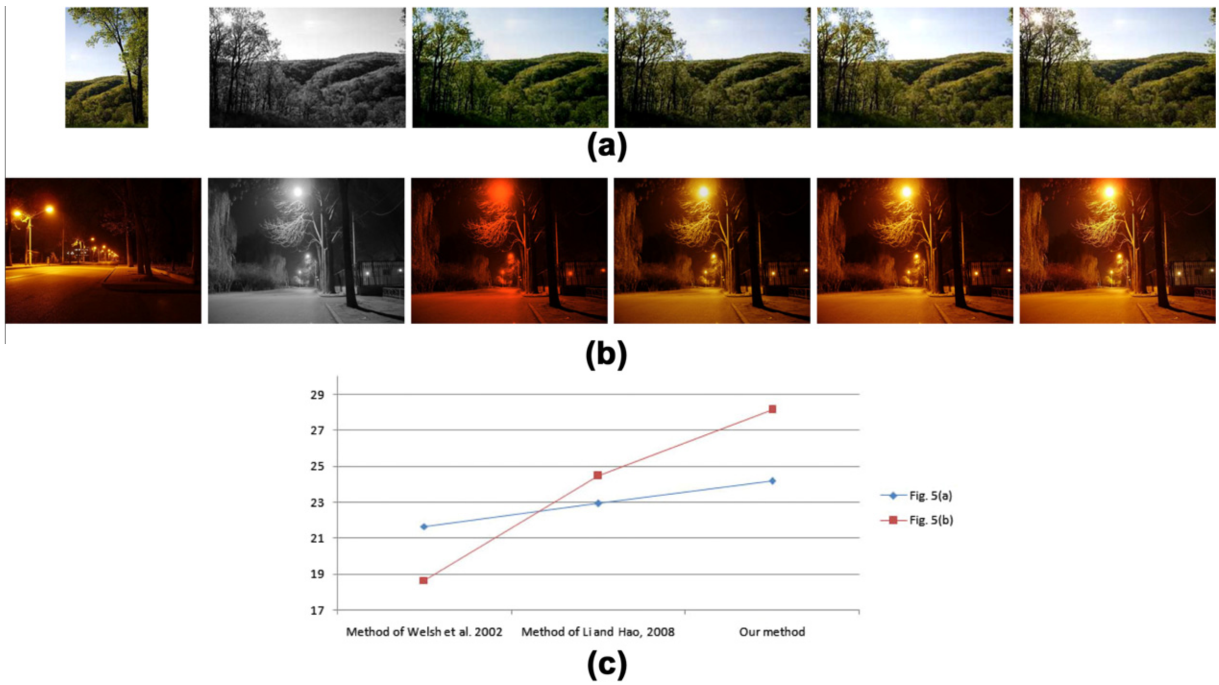

Fig. 5. Contrast between our colorization result and other methods. (a) and (b) From left to right: source image; grayscale image; result of Welsh et al. (2002); result of Li and Hao (2008); result of our method and the ground truth. (c) PSNR contrast between our method and other methods.

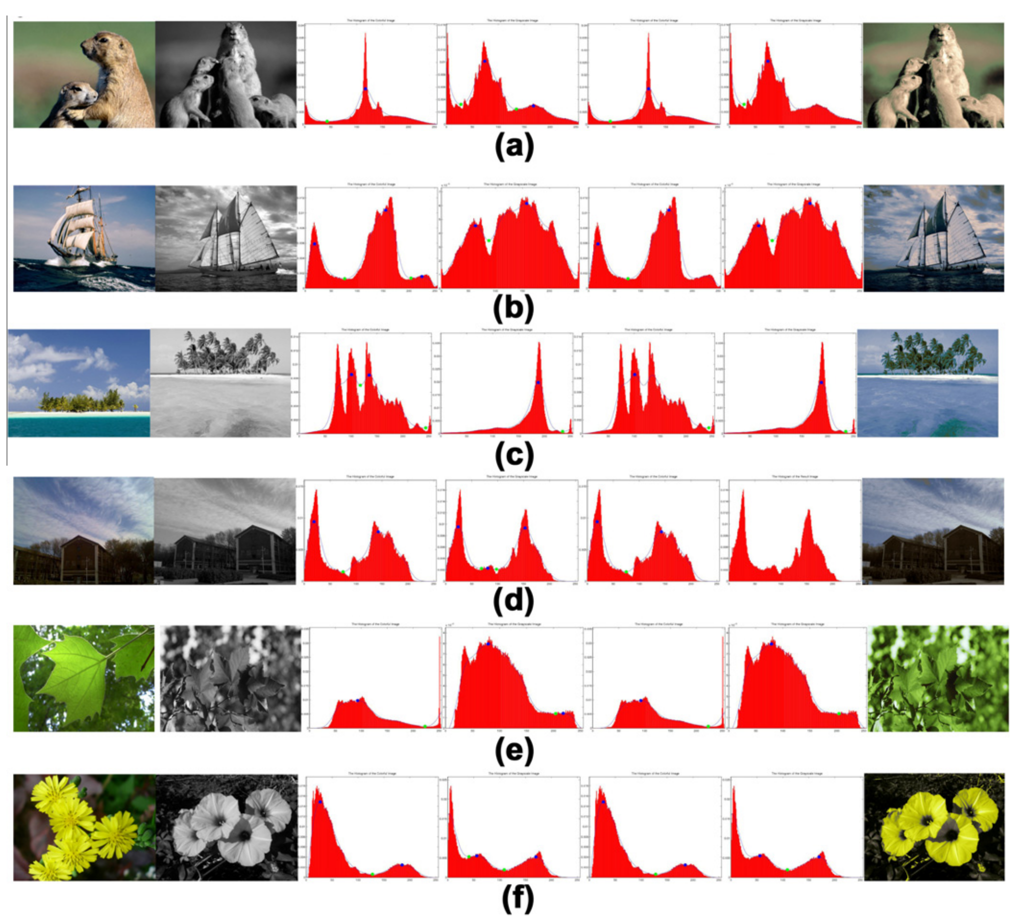

Fig. 6. More colorization results, from left to right: the source image, the grayscale image, the histogram of the source image, the histogram of the grayscale image, the adjusted histogram of the source image, the adjusted histogram of the grayscale image, the colorization result.

information implied by the neighborhood of any particular pixel is insufficient to express how human distinguishes between objects and their textures in an image.

9. Conclusions and future work

In this paper, we presented a novel method to perform automatic grayscale image colorization. First, we applied locally weighted regression to the histograms of the source image and the grayscale image to realize histogram segmentation. Then, the zero-points adjustment algorithm is proposed to implement the pairing procedures between the histograms of the source image and the grayscale image. After the above adjustment, a weighted colorization method is adopted to colorize the grayscale image. We also presented the conception of colorization confidence to qualitatively evaluate the colorization results.

Our method is effective and easy to implement. After the user provides a source image for reference, our system is able to colorize the grayscale image automatically. It can be used to colorize massive images with much less user labor by contrast with conventional colorization methods. As our method is based on the histogram of images, it can well colorize an image with noise in a similar way to an image free of noise. However, if the luminance of the noise is similar to other pixels in the image, mis-matching may occur and thus leading to unwanted colorization results. By providing some reference images for key frames, this method can also be extended to video colorization. This is one of our future works.

Our method can achieve high confident colorization results. However, it is limited to colorize the images without too many texture details. It may fail, for example, to deal with images with strong texture patterns or varied lighting effects like shadows, highlight, etc. Integrating the advanced texture analysis method to our system may be a good solution to this problem. We will further improve our algorithm to make it applicable to more kinds of images.

The only labor of our method is to provide a source image by the user. This may be difficult for some unskilled users, but it provides an easy way to realize the colorization results according to different user's needs. In the future, we will combine our method with the web search techniques to develop a system to meet the needs of more kinds of users. By adopting the modern parallel computational techniques such as GPU (Graphics Processing Units) to achieve real time colorization is also a matter of our future research.

Acknowledgements

We thank the anonymous reviewers for their insightful and valuable comments. The research was supported by National Science Foundation of China Nos. 61170118 and 60803047, the Specialized Research Fund for the Doctoral Program of Higher Education of China No. 200800561045, and the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University No. A1210.

References

Abadpour, A., Kasaei, S., 2007. An efficient PCA-based color transfer method. J. Vision Commun. Image Represent. 18 (1), 15–34.

Charpiat, G., Hofmann, M., Scholkopf, B., 2008. Image colorization via multimodal predictions. In: Proc. 10th European Conf. on Computer Vision (ECCV), pp. 126–139.

Cheng, H., Sun, Y., 2000. A hierarchical approach to color image segmentation using homogeneity. IEEE Trans. Image Process. 9 (12), 2071–2082.

Gonzalez, R.C., Woods, R.E., 1987. Digital Image Processing, second ed. Addison-Wesley Publishing, Reading, Massachusetts.

Heeger, D.J., Bergen, J.R., 1995. Pyramid-based texture analysis/synthesis. In: Proc. SIGGRAPH, pp. 229–238.

Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salessin, D.H., 2001. Image analogies. In: Proc. SIGGRAPH'01, pp. 327–340.

Irony, R., Cohen-or, D., Lischinski, D., 2005. Colorization by example. In: Proc. Rendering Techniques'05, pp. 201–210.

Kass, M., Solomon, J., 2010. Smoothed local histogram filters. ACM Trans. Graphic. 29 (4), 100:1–100:10.

Kawulok, M., Smolka, B., 2010. Competitive image colorization. In: Proc. IEEE 17th Internat. Conf. on Image Processing. pp. 405–408.

Lagodzinski, P., Smolka, B., 2008. Digital image colorization based on probabilistic distance transform. In: Proc. 50th Internat. Symposium ELMAR, pp. 494–498.

Levin, P., Leschinski, D., Weis, Y., 2004. Colorization using optimization. In: Proc. SIGGRAPH'04, pp. 689–694.

Li, J., Hao, P., 2008. Transferring colours to grayscale images by locally linear embedding. In: Proc. British Machine Vision Conf. (BMVC), Leeds, UK, pp. 835–844.

Liu, X., Wan, L., Qu, Y., Wong, T., Lin, L.S., Heng, P.A., 2008. Intrinsic colorization. ACM Trans. Graphic. 27 (5), 1–9.

Luan, Q., Wen, F., Cohen-or, D., Liang, L. et al., 2007. Natural image colorization. In: Proc. Rendering Techniques'07, 309 -320.

Morimoto, Y., Taguchi, Y., Naemura, T., 2009. Automatic colorization of grayscale images using multiple images on the web. In: Proc. SIGGRAPH Posters, Article 32.

Neumann, L., Neumann, A., 2005. Color style transfer techniques using hue, lightness and saturation histogram matching. In: Proc. Computational Aesthetics in Graphics, Visualization and Imaging'05, pp. 111–122.

Nie, D., Ma, L., Xiao, S., Xiao, X., 2005. Grey-scale image colorization by local correlation based optimization algorithm. In: Proc. VISUAL'05, pp. 13–23.

Nie, D., Ma, Q., Ma, L., Xiao, S., 2007. Optimization based grayscale image colorization. Pattern Recognition Lett. 28 (12), 1445–1451.

Pouli, T., Reinhard, D.E., 2010. Progressive histogram reshaping for creative color transfer and tone reproduction. In: Proc. NPAR'10, pp. 81–90.

Qu, Y., Wong, T., Heng, P.A., 2006. Manga colorization. ACM Trans. Graphic. 25 (3), 1214–1220.

Quang, M., Kang, H., Le, T.M., 2010. Image and video colorization using vector-valued reproducing kernel Hilbert spaces. J. Math. Imag. Vision 37 (1), 49–65.

Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P., 2001. Color transfer between images. IEEE Comput. Graph. 21 (5), 34–41.

Senanayake, C.R., Alexander, D.C., 2007. Color transfer by feature based histogram registration. In: Proc. British Machine Vision Conf.

Sýkora D., Buriánek J., Žára J., 2003. Segmentation of black and white cartoons. In: Proc. Spring Conf. on Computer Graphics'03, pp. 245–254.

Welsh, T., Ashikhmin, M., Mueller, K., 2002. Transferring color to grayscale images. In: Proc. SIGGRAPH'02, pp. 277–280.

Xiang, Y., Zou, B., Li, H., 2009. Selective color transfer with multi-source images. Pattern Recognition Lett. 30 (7), 682–689.

Xiao, X., Ma, L., 2009. Gradient-preserving color transfer. Comput. Graph. Forum 28 (7), 1879–1886.

Yatziv, L., Sapiro, G., 2006. Fast Image and video colorization using chrominance blending. IEEE Trans. Image Process. 15 (5), 1120–1129.