Preserving Maximum Color Contrast in Generation of Gray Images

Alex Yong-Sang Chia[1], Keita Yaegashi[2](✉), and Soh Masuko[2]

[1] Rakuten Institute of Technology, Singapore, Singapore
alex.a.chia@rakuten.com
[2] Rakuten Institute of Technology, Tokyo, Japan
{keita.yaegashi,so.masuko}@rakuten.com

Abstract. We propose a method to preserve maximum color contrast when converting a color image to its gray representation. Specifically, we aim to preserve color contrast in the color image as gray contrast in the gray image. Given a color image, we first extract unique colors of the image through robust clustering for its color values. We tailor a non-linear decolorization function that preserves the maximum contrast in the gray image on the basis of the color contrast between the unique colors. A key contribution of our method is the proposal of a color-gray feature that tightly couples color contrast information with gray contrast information. We compute the optimal color-gray feature, and focus the search for a decolorization function on generating a color-gray feature that is most similar to the optimal one. This decolorization function is then used to convert the color image to its gray representation. Our experiments and a user study demonstrate the superior performance of this method in comparison with current state-of-the-art techniques.

Keywords: Image processing · Image decolorization · Feature representation · Coarse-to-fine search

1 Introduction

Image decolorization refers to the process of converting a color image to its gray representation. This conversion is important in applications such as grayscale printing, single channel image/video processing, and image rendering for display on monochromatic devices such as e-book readers. Image decolorization involves dimension reduction, which inevitably results in information loss in the gray image. Therefore, the goal in image decolorization is to retain as much of the appearance of the color image as possible in the gray image. In this work, we aim to generate a gray image in such a way that the maximum of color contrast visible in the color image is retained as gray contrast in the gray image. This ensures different colored patches (both connected and non-connected) of the color image can be distinguished as different gray patches in the gray image. Given that a color image can have arbitrary colors that are randomly distributed across the

© Springer International Publishing Switzerland 2015
A. Fred et al. (Eds.): ICPRAM 2014, LNCS 9443, pp. 252–262, 2015.
DOI: 10.1007/978-3-319-25530-9_17

image, decolorization that focuses on preserving maximum contrast is a very difficult image processing problem.

A key contribution of this paper is the proposal of a novel color-gray feature, which tightly couples color contrast information in a color image with gray contrast information in its corresponding gray image. This provides several unique advantages. First, by encapsulating both color and gray contrast information in a single representation, it allows us to directly evaluate the quality of a gray image based on information available in its color image. More importantly, it provides a convenient means of defining an optimal feature that represents the maximum preservation of color contrast in the form of gray contrast in the gray image. Consequently, by searching for a color-gray feature that is most similar to the optimal feature, we can compute a gray image in which different colored regions in the color image are also distinguishable in the gray image.

We employ a non-linear decolorization function to convert a color image to a gray image. The use of a non-linear function increases the search space for the optimal gray image. To reduce computation cost, we adopt a coarse-to-fine search strategy that quickly eliminates unsuitable parameters of the decolorization function to hone in on the optimal parameter values. We show, through experimental comparisons, that the proposed decolorization method outperforms current state-of-the-art methods.

This paper is organized as follows. Immediately below, we discuss related work. We present our decolorization method in Sect. 2. Experimental evaluations against existing methods are detailed in Sect. 3. We offer our conclusions in Sect. 4.

1.1 Related Work

Traditional decolorization methods apply a weighted sum to each of the color planes to compute a gray value for each pixel. For example, MATLAB [1] eliminates the hue and saturation components from a color pixel, and retains the luminance component as the gray value for each pixel. Neumann et al. [2] conducted a large-scale user study to identify the general set of parameters that perform best on most images, and used these parameters to design their decolorization function. Their methods support very fast computation of gray values, in which the computational complexity is $O(1)$. However, given that the decolorization function is not tailored to the input image, these methods rarely maintain the maximum amount of information from the color image.

Modern approaches to convert a color image to its gray representation tailor a decolorization function to the color image. Such approaches can be classified into two main categories: local mapping and global mapping. In local mapping approaches, the decolorization function applies different color-to-gray mappings to image pixels based on their spatial positions. For example, Bala and Eschbach [3] enhanced color edges by adding high frequency components of chromaticity to the luminance component of a gray image. Smith et al. [4] used a local mapping step to map color values to gray values, and utilized a local sharpening step to further enhance the gray image. While such methods can improve the perceptual quality of the gray representation, a weakness of these methods is that they can distort the appearance of uniform color regions. This may result in haloing artifacts.

Global mapping methods use a decolorization function that applies the same color-to-gray mapping to all image pixels. Rasche et al. [5] proposed an objective function that addresses both the need to maintain image contrast and the need for consistency of the luminance channel. In their method, a constrained multivariate optimization framework is used to find the gray image that optimizes the objective function. Gooch et al. [6] constrained their optimization to neighboring pixel pairs, seeking to preserve color contrast between pairs of image pixels. Kim et al. [7] developed a fast decolorization method that seeks to preserve image feature discriminability and reasonable color ordering. Their method is based on the observation that more saturated colors are perceived to be brighter than their luminance values would suggest. Recently, Lu et al. [8] proposed a method that first defines a bimodal objective function, then uses a discrete optimization framework to find a gray image that preserves color contrast.

One weakness of the methods proposed by Gooch et al. [6], Kim et al. [7], and Lu et al. [8] is that they optimize contrast between neighboring connected pixel pairs and do not consider color/gray differences between non-connected pixels. As a result, different colored regions that are non-connected may be mapped to similar gray values. This results in the loss of appearance information in the gray image. Our framework considers both connected and non-connected pixel pairs to find the optimal decolorization function of an image, and thus does not suffer from this shortcoming.

2 Our Approach

Figure 1 outlines our method, which consists of four main modules. We first extract unique colors from a given a color image. Thereafter, we compute the corresponding gray values of these color values using a currently considered decolorization function. Based on these color and gray values, we compute a color-gray feature to encapsulate the color and gray contrast information in a single representation. The best possible color-gray feature is computed, and we evaluate the quality of the currently considered decolorization function by comparing its color-gray feature with the optimal one. Then, a coarse-to-fine

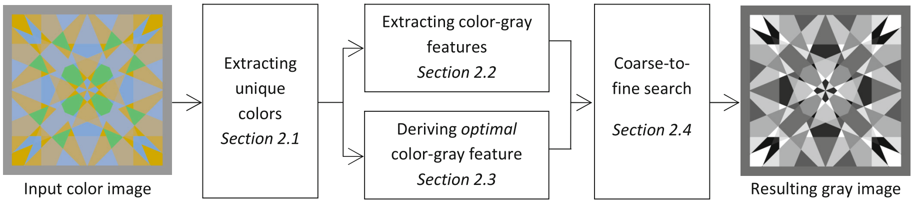

Fig. 1. Overview of our decolorization method.

search strategy is employed to search for a decolorization function that provides the color-gray feature that is the closest fit to the optimal feature. Our method explicitly drives the search toward the gray image that preserves the maximum color contrast in the form of gray contrast. The best-fit function is then used to convert the color image into its gray representation. We elaborate on these modules in the following subsections.

2.1 Extracting Unique Colors

Given an input color image, we first extract its unique color values by applying the robust mean shift clustering method proposed by Cheng [9] to its color values. Let $\{c_i\}$ denote the set of clusters formed. We do not remove weakly populated clusters, but instead consider all mean shift clusters for subsequent processing. Consequently, the color clusters do not represent only the dominant colors of the image; collectively, they represent all the unique colors of the image.

We can highlight three advantages of using these unique colors in the decolorization framework. First, use of these unique colors keeps the computational cost of our method lower than the computational cost of methods that operate on a per-pixel basis, since the number of unique colors is typically much fewer than the number of image pixels. Second, as the clusters represent all unique colors that are extracted from the image, our method does not concentrate the color-to-gray optimization on only the dominant color space of the image but instead on the entire color space represented by the image. Third, and most important, these color clusters do not include spatial information of pixels. By focusing the search for an optimal decolorization function across these color clusters, our method optimizes on a global basis rather than a local neighborhood basis. This ensures that non-connected colored regions map to distinguishable gray regions.

2.2 Extracting Color-Gray Features

Before we describe the computation for a color-gray feature, we must discuss the non-linear decolorization function that we adopt to convert a color image to its gray counterpart. Given a pixel p of a color image, we define its red, green, and blue components as $\{p_r, p_g, p_b\}$ in which the values vary in the range of 0 to 1. p_{gray} denotes the gray value of pixel p. We compute p_{gray} by the multivariate non-linear function:

$$p_{\text{gray}} = ((p_r)^x \times w_r + (p_g)^y \times w_g + (p_b)^z \times w_b)^{\frac{3}{x+y+z}} , \tag{1}$$

where $\{w_r, w_g, w_b\}$ are weight values and $\{x, y, z\}$ are power values that correspond, respectively, to the red, green, and blue components. The non-linear function increases the search space (and affords significant flexibility) to find an optimal gray representation. Our aim is to find the set of weight and power values that results in maximum retention of color contrast in the form of gray contrast in the image.

To compute a color-gray feature that tightly couples color contrast to gray contrast, our process is as follows. For each cluster c_i obtained in Sect. 2.1, we first compute mean color value from all pixels within the cluster. We compute the Euclidean distance between the mean color values of all clusters, and collect the color distances into a distance matrix Φ. Additionally, for each cluster, we compute the gray values for all pixels within the cluster based on a currently considered set of weight $\{w_r, w_g, w_b\}$ and power $\{x, y, z\}$ values using Eq. (1). We use these gray values to compute an average gray value for the cluster. Similarly, we compute the Euclidean distances between the gray values of all clusters, and collect these distances into a matrix Γ. Implicitly, each value in $\Phi(i)$ reflects the color contrast between two color clusters whose gray contrast is given in $\Phi(i)$.

Let F denote a color-gray feature, where $F(j)$ denotes the value at the j^{th} dimension of the feature. Each dimension corresponds to a gray-distance interval $[a, b]$. We define set $\Psi_{(a,b)}$ to be the set of color-distance values whose gray distance is within the interval $[a, b]$. Mathematically, this is represented as follows:

$$\Psi_{(a,b)} = \Phi\left(\phi_{(a,b)}\right), \tag{2}$$

where set $\phi_{(a,b)}$ is the set of matrix indices whose gray distance is within the interval $[a, b]$:

$$\phi_{(a,b)} = \cup k, \quad \forall k,\ a \leq \Gamma(k) \leq b. \tag{3}$$

We compute $F(j)$ as

$$F(j) = \max\left(\Psi_{(a,b)}\right). \tag{4}$$

Figure 2 shows a toy example for computing a feature value $F(j)$. Here, three color clusters are considered, where the clusters are depicted as the color patches in the color-distance matrix Φ given in Fig. 2(a). We show the corresponding gray patches and the gray-distance matrix Γ in Fig. 2(b). Consider the computation of $F(8)$, which corresponds to gray-distance interval $[0.7, 0.8]$. We show all entries in G that belong to this interval by the blue outlines in Fig. 2(b), and the color distances that are considered by the red outlines in Fig. 2(a). Maximum color distance within this gray-distance interval is 0.47, and thus, this is the feature value of $F(8)$ (as shown in Fig. 2(c)).

We note here the following points. First, a feature dimension corresponds to a gray-distance interval, and a feature value corresponds to a color-distance value. Therefore, the proposed color-gray feature directly incorporates both color and gray information in its representation. More importantly, each feature value indicates the maximum color contrast of the color image that is represented within the currently considered gray-contrast interval, and indicates the importance of the gray-contrast interval. This information enables us to compute an optimal feature that retains maximum color contrast of the color image in the form of gray contrast, as described below.

2.3 Deriving Optimal Color-Gray Feature

We seek color-gray features in which larger feature values are present in the dimensions of the feature which are furthest to the right. Intuitively, feature

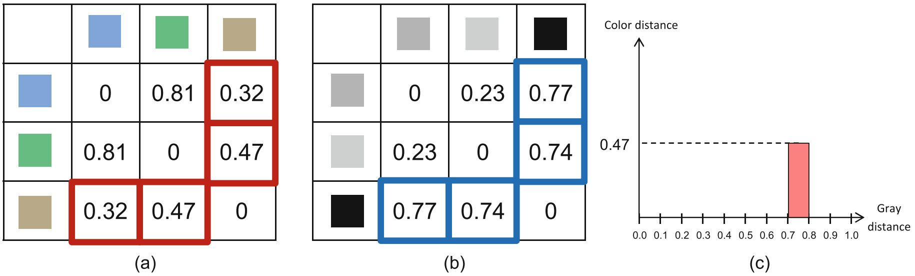

Fig. 2. Toy example for computing value $F(8)$ for three color clusters. (a) Color-distance matrix F for the three color clusters. (b) Corresponding gray-distance matrix G. (c) Feature value $F(8)$ computed at intervals $[0.7, 0.8]$. Entries in F and G used to compute $F(8)$ are represented by the red and blue outlines in (a) and (b) respectively (Color figure online).

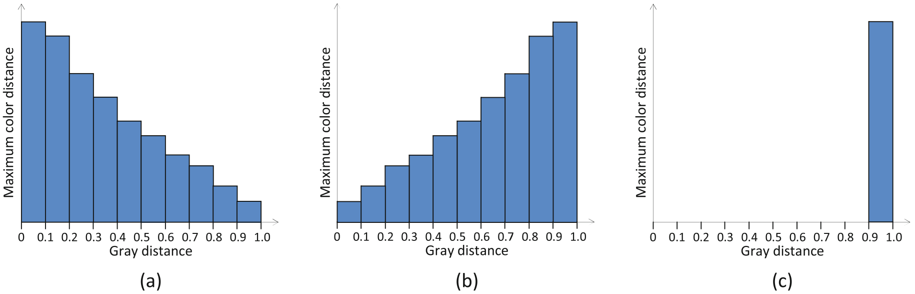

Fig. 3. Pictorial representations of (a) poor, (b) good and (c) optimal color-gray features. See text for details (Color figure online).

dimensions depict the extent clusters $\{c_i\}$ that can be distinguished in the gray space, whereas larger dimensions correspond to a higher gray contrast, and therefore more perceivable differences in the gray space. A feature value $F(j)$ indicates the importance of the feature dimension j in the color space, where larger feature values indicate that clusters exist that are readily distinguishable in the color space. Thus, a color-gray feature that has a heavy right tail corresponds to a color-to-gray mapping in which clusters whose color contrasts are readily distinguished in the color space have gray contrast that is easily perceived in the gray space. Figure 3 provides a pictorial representation of several color-gray features. The feature of Fig. 3(a) implies that clusters that are readily distinguished in the color space (i.e., have high color contrast) are weakly perceived in the gray space. Conversely, the feature in Fig. 3(b) shows that clusters that have sharp color contrast can be readily distinguished in the gray space. This indicates a better gray representation of the color clusters. By extending this reasoning, we can derive the optimal color-gray feature in Fig. 3(c), where regardless of the color differences between the clusters, these clusters have maximum contrast in the gray space.

2.4 Coarse-to-Fine Search Strategy

We compare a color-gray feature, generated by a current set of weight $\{w_r, w_g, w_b\}$ and power $\{x, y, z\}$ values, with the optimal feature in order to evaluate the quality of the set of weight and power values. Our aim is to determine the minimum computational cost of transforming the current color-gray feature to the optimal one. The feature values are normalized to sum to 1, and then we use the earth mover's distance to compare the feature vectors. A naïve method of identifying the best set of weight and power values is to perform iteration over all values, and select the values whose color-gray feature has the minimum earth mover's distance to the optimal feature. This, however, has high computational costs in the order of $O\left(n^6\right)$. Instead, we adopt a fast coarse-to-fine hierarchical search strategy to find the best set of weight and power values. Specifically, we first search across coarse ranges of the weight values, and identify a seed weight value $\left\{w_r^*, w_g^*, w_b^*\right\}$ that has the least earth mover's distance from the optimal color-gray feature. Thereafter, we search at a finer scale in the neighborhood of $\left\{w_r^*, w_g^*, w_b^*\right\}$. To ensure that we do not get trapped in a local minimum, we retain three sets of seed weight values that have the smallest earth mover's distances, and conduct the fine search across the neighborhoods of these sets. At the termination of the fine search, we identify the weight values that have the least distance from the optimal color-gray feature and search across various ranges of the power values of these weight values. The decoupling of the search for the power values from the weight values, together with the coarse-to-fine search strategy, makes our method substantially faster than the brute force method. In this paper, a coarse search is conducted in the range $\{0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0\}$ and a fine search at an offset of $\{-0.10,\ -0.05,\ 0,\ +0.05,\ 0.10\}$ from three sets of seed weight values. During computation, we ignore a set of weight values if the sum of the weights exceeds 1. The search for the power values is in the range $\{0.25,\ 0.5,\ 0.75,\ 1.0\}$. The proposed method searches over a maximum of 435 different value settings, as opposed to $47,439$ settings using the naïve search method. This yields more than 100× increase in speed with our method, as compared to the naïve method.

3 Results

We compare our method with MATLAB's *rgb2gray* function, and the recent state-of-the-art methods of Lu et al. [8], Rasche et al. [5], and Smith et al. [4]. For all experiments, we evaluate the methods using the publicly available color-to-gray benchmark dataset of Čadík [10], which comprises 24 images. Image decolorization by our method on all test images is achieved with the same set of parameter settings, and takes under one minute per image on unoptimized codes. We construct color-gray features using 20 equally spaced gray intervals. Three sets of seed weight values are computed from coarse intervals of $\{0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0\}$. These seed weight values are then used to initialize a fine search at an offset of $\{-0.10,\ -0.05,\ 0,\ +0.05,\ 0.10\}$. We find the best power values by searching across values $\{0.25,\ 0.5,\ 0.75,\ 1.0\}$.

Figure 4 show decolorization results obtained by the proposed and comparison methods across various synthetic and real images. We show the input color images in the first column. Gray images computed with MATLAB's *rgb2gray* function and the methods of Lu et al. [8], Rasche et al. [5], and Smith et al. [4] are shown in the second to fifth columns, respectively. Gray images obtained by our method are shown in the final column. As can be seen in the figure, the proposed method provides perceptually more meaningful representation of the

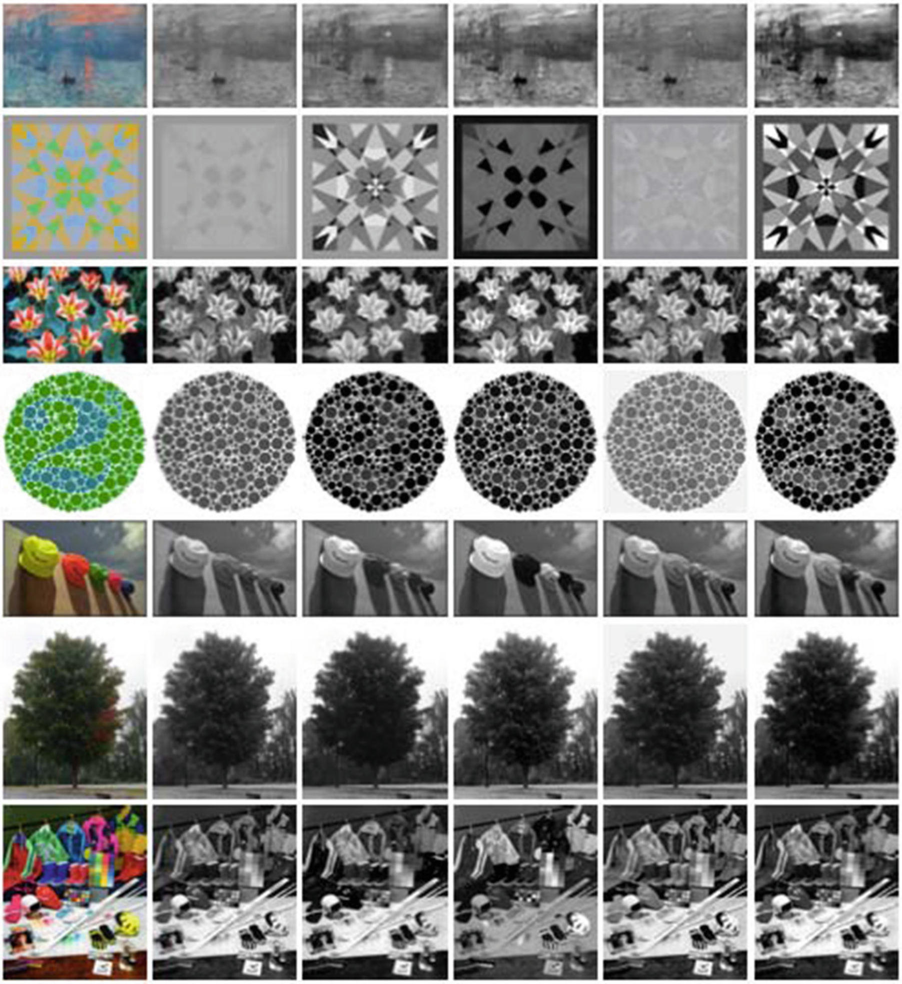

Fig. 4. Decolorization results. First column shows reference color image. Gray images obtained with MATLAB's rgb2gray function, the method of Lu et al. [8], the method of Rasche et al. [5], and the method of Smith et al. [4] are shown in the second to fifth columns, respectively. The final column shows gray images obtained with our method. This figure is best viewed with magnification (Color figure online).

color images, where color contrast present in the images is well preserved as gray contrast. For example, consider the synthetic image shown in the first row of Fig. 4. We note that our method resulted in superior representation, with the contrast between the red sun and the background being better maintained in our gray image than in the gray images of the comparison methods. Additionally, contrast in the fine scale details in the middle-right portion of the image are well preserved using our method. Decolorization results on real images also bear out the superior performance of our method. For example, consider the real image shown in the fifth row of Fig. 4. Our gray representation of the hats in the figure renders them distinguishable in the gray image, and is an improved representation as compared with representations produced by the other methods. An interesting example is shown in the sixth row of the figure, in which the color image shows a green tree with small red patches on its right side. Our method is able to generate a gray image in which red and green patches in the color image can be distinguished as different gray patches in the gray image. In contrast, these patches are indistinguishable in gray images produced by the comparison methods.

Quantitative Evaluation. We use the color contrast preserving ratio (CCPR) proposed by Lu et al. [8] to quantitatively compare our method with other methods. Color difference between pixel p and q is calculated as follows:

$$\delta_{p,q} = \sqrt{(p_r - q_r)^2 + (p_g - q_g)^2 + (p_b - q_b)^2}, \tag{5}$$

and CCPR is defined as follows:

$$\text{CCPR} = \frac{\#\{(p,q)|(p,q) \in \Omega,\ |p_{\text{gray}} - q_{\text{gray}}| \geq \tau\}}{\|\Omega\|} \tag{6}$$

where Ω is the set containing all neighboring pixel pairs with their original color difference between pixel p and q, which is computed as $\delta_{p,q} \geq \tau$. $\|\Omega\|$ is the number of pixel pairs in Ω. $\#\{(p,q)|(p,q) \in \Omega,\ |p_{\text{gray}} - q_{\text{gray}}| \geq \tau\}$ is the number of pixel pairs that are still distinctive after decolorization.

This measurement evaluates the ability of the methods to preserve contrast. Larger ratios indicate better ability to preserve color contrast in the color image as gray contrast in the gray image. We calculate the average CCPR for the whole dataset by varying τ from 1 to 15. We report the mean and standard deviation of the ratios in Table 1. The quantities indicate our method can satisfactorily preserve color distinctiveness and perform better than the other methods in preserving contrast. A t-test shows the comparison results to be statistically significant ($\rho < 10^{-7}$).

User Study. We perform a user study to qualitatively evaluate our method. We engaged 60 subjects with normal vision (whether natural or corrected) for the study. We show each subject 24 sets of images. Each set consists of a reference color image and decolorized results obtained by the proposed and comparison methods. To avoid bias, we randomly jumble the ordering of the

Table 1. Quantitative evaluation using color contrast preserving ratio (CCPR) proposed by Lu et al. [8].

	Mean	Std
MATLAB *rgb2gray*	0.7134	0.2872
Lu et al. [8]	0.8213	0.1595
Rasche et al. [5]	0.7442	0.2565
Smith et al. [4]	0.8212	0.2708
Proposed method	0.8497	0.1621

gray images before presenting them to the subjects. Subjects are instructed to identify the gray image that best represents color patches in the color image as different gray patches in the gray image. Overall, the subjects identify gray images produced by our method as the best representations 27.2 % of the time. This compares favorably to selection of the comparison methods, which are as follows: 16.1 % MATLAB, 18.5 % Lu et al. [8], 19.3 % Rasche et al. [5], and 18.9 % Smith et al. [4]. A t-test shows the comparison results to be statistically significant ($\rho < 2.0 \times 10^{-3}$).

4 Discussion

We propose an image decolorization method that retains maximum color contrast in a color image as gray contrast in the corresponding gray image. To this end, we propose a novel color-gray feature that couples color contrast and gray contrast information. This feature provides a unique ability to directly evaluate the quality of a gray image based on information available in the color image. More importantly, it provides a mechanism to drive our search to find the color-to-gray decolorization function that best preserves contrast in the gray image. A non-linear decolorization function is employed to convert a color image to its gray representation, and we reduce computational cost by using a coarse-to-fine search strategy. Experimental comparisons and a user study show the superior effectiveness of our approach. For future work, we are interested in extending our method to decolorize movie frames. We will exploit spatial and temporal cues to ensure coherence in gray representation is maintained across different movie frames.

References

1. The MathWorks, I.: MATLAB version 7.10.0 (r2010a) (2010). http://www.mathworks.com/products/matlab/
2. Neumann, L., Čadík, M., Nemcsics, A.: An efficient perception-based adaptive color to gray transformation. In: Proceedings of the Third Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging, Eurographics Association, pp. 73–80 (2007)

3. Bala, R., Eschbach, R.: Spatial color-to-grayscale transform preserving chrominance edge information. In: Color and Imaging Conference, vol. 2004, Society for Imaging Science and Technology, pp. 82–86 (2004)
4. Smith, K., Landes, P.E., Thollot, J., Myszkowski, K.: Apparent greyscale: a simple and fast conversion to perceptually accurate images and video. Comput. Graph. Forum **27**, 193–200 (2008)
5. Rasche, K., Geist, R., Westall, J.: Re-coloring images for gamuts of lower dimension. Comput. Graph. Forum **24**, 423–432 (2005)
6. Gooch, A.A., Olsen, S.C., Tumblin, J., Gooch, B.: Color2gray: salience-preserving color removal. ACM Trans. Graph. (TOG) **24**, 634–639 (2005)
7. Kim, Y., Jang, C., Demouth, J., Lee, S.: Robust color-to-gray via nonlinear global mapping. ACM Trans. Graph. (TOG) **28**, 161 (2009)
8. Lu, C., Xu, L., Jia, J.: Real-time contrast preserving decolorization. In: SIGGRAPH Asia 2012 Technical Briefs, p. 34. ACM (2012)
9. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**, 790–799 (1995)
10. Čadík, M.: Perceptual evaluation of color-to-grayscale image conversions. Comput. Graph. Forum **27**, 1745–1754 (2008)