

Color Consistency for Photo Collections Without Gamut Problems

Qi-Chong Tian^(✉) and Laurent D. Cohen

Université Paris-Dauphine, PSL Research University, CNRS, UMR 7534,
CEREMADE, 75016 Paris, France
{tian,cohen}@ceremade.dauphine.fr

Abstract. In this paper, we present a color consistency technique in order to make images in the same collection share the same color style and to avoid gamut problems. Some previous methods define simple global parameter-based models and use optimizing algorithms to obtain the unknown parameters, which usually cause gamut problems in bright and dark regions. Our method is based on the range-preserving histogram specification and can enforce images to share the same color style, without resulting in gamut problems. We divide the input images into two sets having respectively high visual quality and low visual quality. The high visual quality images are used to make color balance. And then the low visual quality images are color transferred using the previous corrected high quality images. Our experiments indicate that such histogram-based color correction method is better than the compared algorithm.

Keywords: Color consistency · Color correction · Histogram specification · Gamut preserving · Image quality assessment

1 Introduction

With the popularity of digital cameras, online photo albums and social networks, we can easily find thousands of photos of the same scenes, especially for the famous landmarks. And these kinds of photo collections have been applied in high level computer vision problems, including visual photo tourism [1], scene completion [2], landmark recognition [3], content-based image retrieval [4], photo stitching [5], and time-lapse [6]. However, these online images are usually taken with different capture devices [7], different illuminations, and different views, which are the reasons for the different color styles in the same photo collection.

In order to obtain better results of the above mentioned interesting computer vision applications, the color consistency or color correction is an essential pre-processing procedure before implementing the main algorithms. The most simple color correction method is the auto color adjustment technique, which is implemented in most commercial photo editing tools. However, this kind of method makes the color correction for each image independently, not considering the color distribution relations among all images in the same collection.

Another kind of method is manually adjusting all images by professional photographers, which requires a significant amount of processing time for a large number of photos.

Recently, Park et al. [8] proposed a method to optimize color consistency in community photo collections depicting the same scenes. Their method is based on the assumption of a color correction model, which is a simple combination of white balance and gamma correction. Park et al.’s method is globally making color mapping in the three color channels respectively, and may cause gamut problems, since the dark pixel values are easily saturated to 0 and the bright pixel values are easily saturated to 255 in this simple global parameter-based model. Another disadvantage of their algorithm is that the final results are highly depending on the color distribution of all images. In other words, some low quality images with bad color distribution will affect the quality of other images in the same collection.

In order to avoid gamut problems [9, 10] and reduce the effects from bad quality images, we present an automatic color consistency algorithm for photo collections based on image quality assessment and range-preserving histogram specification. In our method, input images are divided into two sets, namely high quality image set and low quality image set. And then the images having high quality will be used to produce corresponding color balanced images. At last, the images classified in the low quality image set will be color transferred with the corrected high quality images as the reference. These three stages can obtain color consistency among the whole set of images and not cause the disadvantages existing in Park et al.’s method.

The rest of this paper is organized as follows. The relevant previous work is summarized in Sect. 2. The overview of the proposed algorithm is described in Sect. 3 and the detailed description is given in Sect. 4. The experiments and comparisons are shown in Sect. 5. Finally, we give the conclusion and discuss the future work in Sect. 6.

2 Related Work

In this section, we summarize the previous related work and categorize these methods by the number of input images. Color image enhancement is an ancient image processing technique, which is normally applied to a single image based on the statistic information of the input image or some specialized filters. Color transfer is a technique of transferring the color style of a reference image to a test image. Color balance for image collections is a new and interesting problem, which can enforce thousands of images to share the same color style.

2.1 Color Enhancement for a Single Image

Color image enhancement is an ancient and challenging task in image processing. Histogram equalization [11] is a useful method for image enhancement. Sapiro and Caselles [12] proposed a PDE-based image histogram modification method,

which was further extended to color image enhancement by many researchers [13–17]. The automatic color enhancement (ACE) algorithm [18] is a popular method, which has been used in many photo editing software. Recently, Yuan and Sun [19] proposed an automatic exposure correction method, which was based on region-level exposure evaluation and detail-preserving S-curve adjustment. This kind of algorithm operates on each image independently, so they can not obtain color consistency among all images in the same collection.

2.2 Color Transfer Between Two Images

Color transfer between two images has received significant attention in recent years. The goal of this kind of algorithm is transferring the color style of a reference image to a test image, which can enforce the corresponding two images to share the same color style. Reinhard et al. [20] proposed a simple statistics-based algorithm to transfer colors between two images, which was also extended by many researchers. Papadakis et al. [21] proposed a variational model for color image histogram transfer, which used the energy functional minimization to obtain the goal of transferring the image color style and maintaining the image geometry. Hristova et al. [22] presented a style-aware robust color transfer method, which was based on the style feature clustering and the local chromatic adaptation transform. Hwang et al. [23] proposed a geometry-based color transfer algorithm using probabilistic moving least squares. A more detailed review of this kind of algorithm can be seen in [24].

2.3 Color Balance for Image Collections

Automatic color consistency for photo collections firstly obtained attention in HaCohen et al. [25]. Their method used the non-rigid dense correspondence algorithm [26] to obtain matched features, and then constructed a graph with edges linking image pairs sharing the similar contents. Color consistency was achieved by globally optimizing three piece-wise quadratic curves over the whole graph, which can minimize the color difference between all correspondences. However, their method is highly depending on dense and accurate correspondences (high computational complexity), which makes this algorithm unsuitable for a large number of images in photo collections.

A more efficient and robust color consistency algorithm for photo collections was proposed by Park et al. [8]. They firstly recovered sparse point correspondences in the same image collection, and then constructed a simple color correction model which can be considered the combination of white balance and gamma correction. Then a robust low-rank matrix factorization method was used to estimate the unknown parameters of the defined model. However, this correction model does not consider gamut problems and negative effects resulting from low quality images in the same collection.

3 Overview

Given an input photo collection as a sequence of images $I = \{I_1, I_2, I_3, \dots, I_n\}$ which describe the same scene with different views, different capture devices or different illustrations, our goal is to make these images share the same color style and to be color consistent in the shared contents.

Our method is inspired by [8], which has given a whole framework to minimizing color difference among all input images. However, we do not adopt the simple global parameter-based model to correct colors, which may result in gamut problems. The contributions of our work are classifying input images based on image quality metrics and finding suitable target histograms for the range-preserving histogram specification.

Figure 1 illustrates the overall framework of our color consistency algorithm, which consists of three main stages.

1. Image quality assessment: use non-reference measures to assess the image visual quality, and then construct two image sets having respectively high quality and low quality.

2. Color balance: use matched points between images having high quality to construct correspondences, and then obtain early corrected images with linear transforms. These early images can produce the target histograms. At last, the range-preserving histogram specification is applied to each image.

3. Color transfer: use matched points between low quality images and corrected high quality images to construct correspondences, and then obtain final corrected images with linear transforms (can produce target histograms) and the range-preserving histogram specification.

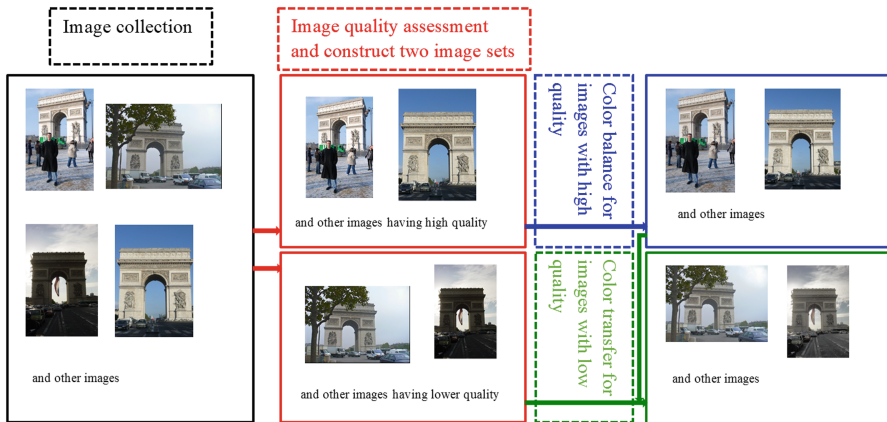


Fig. 1. The overall framework of the proposed method

4 Color Consistency without Gamut Problems

4.1 Image Quality Assessment by Non-reference Measures

As discussed in Sect. 1, the results of Park et al.'s method [8] depend on the color distribution of all images in the same collection. This kind of algorithm may obtain some bad results when there are many low quality images in the same image collection. So we firstly make quality assessment for input images, which can sort the images by their visual quality. Then we make color balance for the images ranking high quality. At last, this kind of color style obtained from high visual quality images will be propagated to the images which are ranked lower visual quality in the assessment procedure. This kind of processing can reduce bad effects from lower visual quality images.

Much effort [27, 28] has been made in recent years to design non-reference image quality assessment metrics. Panetta et al. [27] proposed a non-reference color image quality measurement combining the contrast, colorfulness and sharpness metrics. In other words, this measurement is a linear weighting of three terms assessing the contrast, color and edge information respectively. We simplify this assessment method and adopt the contrast and the sharpness metrics in our method. The measurement IQA (Image Quality Assessment) is written in Eq. (1).

$$IQA = w_1 * Contrast + w_2 * Sharpness, \quad (1)$$

where w_1 is the weighting of the contrast measure, w_2 is the weighting of the sharpness measure, *Contrast* is defined in Eq. (2), *Sharpness* is defined in Eq. (3).

$$Contrast = AME(I_Y)$$

$$AME(I_Y) = \frac{1}{k_1 * k_2} \sum_{l=1}^{k_1} \sum_{k=1}^{k_2} \left(\log \left(\frac{I_Y(max, k, l) + I_Y(min, k, l)}{I_Y(max, k, l) - I_Y(min, k, l)} \right) \right)^{-0.5}, \quad (2)$$

where AME is the Michelson-Law measure of enhancement [29], I_Y is the values of Y channel of the input image in the YCbCr color space, the image is divided into $k_1 * k_2$ blocks, $I_Y(min, k, l)$ is the minimum of the block (k, l) in I_Y , $I_Y(max, k, l)$ is the maximum of the block (k, l) in I_Y , k and l are the row and column index of blocks in the image.

$$Sharpness = \sum_{c=1}^3 \lambda_c EME(GrayEdge_c) \quad (3)$$

$$EME(GrayEdge) = \frac{2}{k_1 * k_2} \sum_{l=1}^{k_1} \sum_{k=1}^{k_2} \log \left(\frac{GrayEdge(max, k, l)}{GrayEdge(min, k, l)} \right),$$

where c is the index of the color channel, EME is the measure of enhancement [30], $GrayEdge_c$ is the edge map of the corresponding color channel, λ_c is the weighting of each color channel, the image is divided into $k_1 * k_2$ blocks.

$GrayEdge(min, k, l)$ and $GrayEdge(max, k, l)$ are respectively the minimum and maximum of the block (k, l) in $GrayEdge$, k and l are the row and column index of blocks in the image.

After this kind of quality assessment, we obtain the corresponding visual quality values of each image, which are written by $IQA = \{IQA_1, IQA_2, IQA_3, \dots, IQA_n\}$. Then these images can be sorted by descending the values of IQA . Some best quality images $I_{high} = \{I_{high(1)}, I_{high(2)}, \dots, I_{high(s)}\}$ will be chosen for make color balance in the next stage. s is the total number of images in the set I_{high} . The rest images are labeled by $I_{low} = \{I_{low(1)}, I_{low(2)}, \dots, I_{low(t)}\}$, where $t = n - s$.

4.2 Color Balance Without Gamut Problems

In this stage, the goal is to make color balance for the image set $I_{high} = \{I_{high(1)}, I_{high(2)}, \dots, I_{high(s)}\}$. We adopt the global color correction method, which is robust to feature matching errors and has low computational complexity. We do not define the simple correction models which were defined in [8, 25, 26]. For example, the correction model in [8] is the simulation of white balance and gamma correction, which is depending on only two parameters for each image and may result in gamut problems. Our color balance method consists of two parts, linear transforms to obtain target histograms and range-preserving histogram specifications to produce corrected images. The detailed algorithm is described in **Algorithm 1**.

Algorithm 1. Color Balance without Gamut Problems

- 1: **Input:** $I_{high} = \{I_{high(1)}, I_{high(2)}, \dots, I_{high(s)}\}$.
 - 2: **Extract SIFT features** in $I_{high(i)}$, where $i \in [1, 2, \dots, s]$.
 - 3: **Feature matching** between $I_{high(i)}$ and $I_{high(j)}$, where $i, j \in [1, 2, \dots, s]$ and $i \neq j$, store matching results in a sparse Matrix M_1 with size $k \times s \times 3$, k is the number of matching points in the same scene.
 - 4: **Correction for the sparse Matrix** M_1 : find the non-zero values in each row of M_1 , compute the mean of these non-zero values and replace the original non-zero values, which produces a new sparse Matrix M_2 .
 - 5: **Compute linear transforms** $T(i)$ for each image: $M_2(i) = M_1(i) \bullet T(i)$, the sizes of $M_1(i)$ and $M_2(i)$ both are $k \times 3$, the size of $T(i)$ is 3×3 .
 - 6: **Compute transformed images:** $I_{Trans(i)} = I_{high(i)} \bullet T(i)$.
 - 7: **Histogram Specification** for $I_{high(i)}$ with the histogram of $I_{Trans(i)}$, the result is $I_{Correction(i)}$.
 - 8: **Output:** $I_{Correction(i)}$.
-

For the step 7 in **Algorithm 1**, we combine the hue-preserving color image enhancement framework [9] and the fast exact histogram specification [15, 16], which can avoid gamut problems. Given a color image IM , we firstly get the corresponding intensity image G using Eq. (4). Then the histogram specification will be applied to G , the result is \hat{G} . At last, we use the hue-preserving

enhancement framework to obtain the final corrected color image \hat{IM} , which is computed using the Eq. (5) for the three color channels respectively.

$$G = 0.299 * I_R + 0.587 * I_G + 0.114 * I_B, \quad (4)$$

where I_R, I_G, I_B are respectively the red color channel, green color channel and blue color channel of the corresponding image IM .

$$\begin{aligned} im\hat{(k)} &= \frac{G\hat{(k)}}{G(k)} im(k), & \text{if } \frac{G\hat{(k)}}{G(k)} \leq 1, \\ im\hat{(k)} &= \frac{255 - G\hat{(k)}}{255 - G(k)} (im(k) - G(k)) + G\hat{(k)}, & \text{if } \frac{G\hat{(k)}}{G(k)} > 1, \end{aligned} \quad (5)$$

where im is the corresponding color channel of the image IM , $im\hat{}$ is the corresponding color channel of the final corrected image \hat{IM} , k is the index of pixels in each gray image.

4.3 Color Transfer Without Gamut Problems

In the first stage, we divided the input image collection into two image sets, based on a non-reference image quality assessment method. The first image set has been color balanced in the previous stage. And the other image set having lower quality will be color corrected in this stage. We use the idea of color transfer or color propagation between two images to finish the task at this stage. The detailed algorithm is described in **Algorithm 2**.

Algorithm 2. Color Transfer without Gamut Problems

- 1: **Input:** $I_{low} = \{I_{low(1)}, I_{low(2)}, \dots, I_{low(t)}\}$.
 - 2: **Extract SIFT features** in $I_{low(j)}$, where $j \in [1, 2, \dots, t]$.
 - 3: **Feature matching** between $I_{low(j)}$ and $I_{high(i)}$, where $j \in [1, 2, \dots, t]$, $i \in [1, 2, \dots, s]$, $I_{high(h(j))}$ is the best corresponding image pair of $I_{low(j)}$, the corresponding values are stored in two Matrices M_j and $M_{h(j)}$.
 - 4: **Compute linear transforms** $T(j)$ for each image: $M_{h(j)} = M_j \bullet T(j)$.
 - 5: **Compute transformed images:** $I_{Trans(j)} = I_{low(j)} \bullet T(j)$.
 - 6: **Histogram Specification** for $I_{low(j)}$ with the histogram of $I_{Trans(j)}$, the result is $I_{Correction(j)}$.
 - 7: **Output:** $I_{Correction(j)}$.
-

For the step 6 in **Algorithm 2**, we adopt the same method described in the previous stage, which can avoid gamut problems.

5 Experiments

5.1 Image Datasets

We had run our algorithm and the comparison algorithm [8] on the landmark images download from FLICKR, which were also collected as a public test dataset

on the project website (landmark3d.codeplex.com) [3]. This dataset was popularly used as a benchmark for evaluating and comparing different computer vision algorithms. In our experiments, four landmarks had been chosen to test color consistency algorithms. ARC TRIOMPHE, NOTRE DAME CATHEDRAL, STATUE OF LIBERTY, TREVI FOUNTAIN are the names of these landmarks. For each landmark, we chose 20 images having different color styles, which were captured with different cameras, different illuminations, different views by different photographers. Some examples will be shown in the following part.

5.2 Results

For the comparison algorithm, we use the original codes shared on the author’s homepage and the parameters are the default values. Our method is implemented on the Matlab platform. Figures 2, 3, 4 and 5 show the results of Park et al.’s method [8] and our method. Since there is no efficient object evaluation metric to assess the performance of color consistency algorithms for photo collections, we compare the results with the subject method based on the color distribution and visual quality of corrected images.



Fig. 2. Results on the photo collection ARC TRIOMPHE, only four images are shown. The upper row shows the original images, the middle row shows the results by Park et al.’s method [8], the lower row shows our results. Red and yellow squares indicate gamut problems in Park et al.’s method. (Color figure online)

The results shown in Fig. 2 demonstrate that our approach can enforce the image collection to share a similar color style and not cause gamut problems.

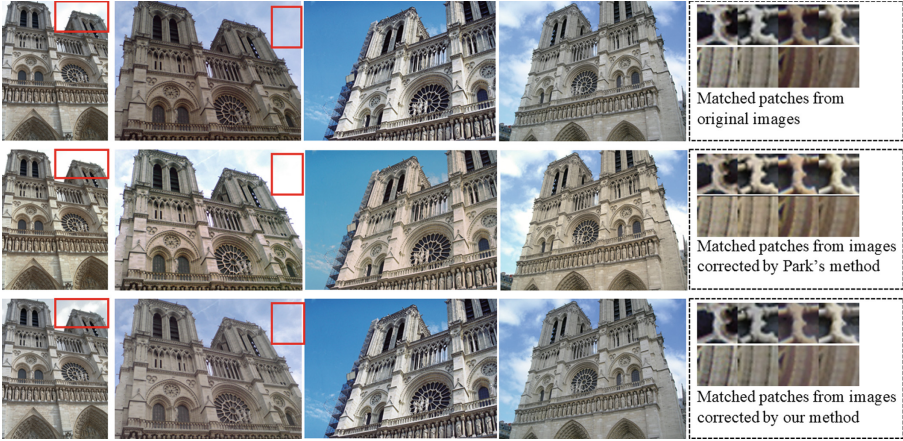


Fig. 3. Results on the photo collection NOTRE DAME CATHEDRAL, only four images are shown. The upper row shows the original images, the middle row shows the results by Park et al.'s method [8], the lower row shows our results. Red squares indicate gamut problems in Park et al.'s method. (Color figure online)



Fig. 4. Results on the photo collection TREVI FOUNTAIN, only four images are shown. The upper row shows the original images, the middle row shows the results by Park et al.'s method [8], the lower row shows our results. Red squares indicate gamut problems in Park et al.'s method. (Color figure online)

The matched patches from the original images and the corrected images can easily describe the color consistency after our method. The yellow squares indicate that the compared algorithm causes gamut problems in the dark regions. In the

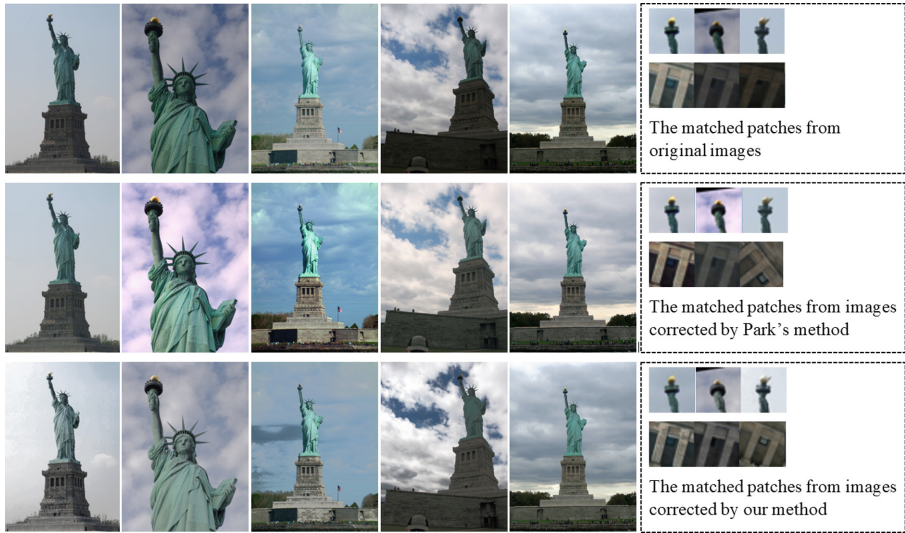


Fig. 5. Results on the photo collection STATUE OF LIBERTY, only five images are shown. The upper row shows the original images, the middle row shows the results by Park et al.'s method [8], the lower row shows our results.

same way, the red squares indicate that it causes gamut problems in the bright regions. Our method do not result in this kind of disadvantage, since we adopt a range preserving exact histogram specification in the proposed algorithm.

In Fig. 3, the results of Park et al.'s method and our approach show these two methods both enforce the images to share a similar color style. However, our method outperforms the compared algorithm on the bright sky regions, which is easy to cause gamut problems in Park et al.'s method. The results shown in Fig. 4 are similar to the results of Fig. 3, which indicates that the two approaches both make color consistency well and Park et al.'s method can not correct pixels in bright and dark regions.

In Fig. 5, it is not easy to compare the results from the whole images on the left of this figure. However, we can know that our results are slightly better than the compared algorithm, from the re-sized matched patches on the right of the figure.

6 Conclusion

In this work, we have proposed a color consistency approach for image collections, based on range-preserving histogram specification. This method can enforce the same image collection to share the same color style and not to cause gamut problems. The main contributions of our work are reducing the effects of bad visual quality images to the whole image collection based on non-reference image

quality assessment method, and finding suitable target histograms for range-preserving histogram specifications. The experiments demonstrate the proposed method outperforms the compared algorithm in terms of color consistency and avoiding gamut problems.

The proposed method is also based on a global model, since the global histogram specification is adopted. Generally speaking, global model may result in some color artifacts when the input images have some bright and dark regions. So we will focus on the local-based model in the future.

References

1. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph. (TOG)* **25**, 835–846 (2006)
2. Hays, J., Efros, A.A.: Scene completion using millions of photographs. *Commun. ACM* **51**, 87–94 (2008)
3. Hao, Q., Cai, R., Li, Z., Zhang, L., Pang, Y., Wu, F.: 3D visual phrases for landmark recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3594–3601 (2012)
4. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: state of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)* **2**, 1–19 (2006)
5. Shan, Q., Curless, B., Furukawa, Y., Hernandez, C., Seitz, S.M.: Photo uncrop. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014. LNCS*, vol. 8694, pp. 16–31. Springer, Heidelberg (2014). doi:[10.1007/978-3-319-10599-4_2](https://doi.org/10.1007/978-3-319-10599-4_2)
6. Martin-Brualla, R., Gallup, D., Seitz, S.M.: Time-lapse mining from internet photos. *ACM Trans. Graph. (TOG)* **34**, Article No. 62 (2015)
7. Faridul, H.S., Stauder, J., Trémeau, A.: Illumination and device invariant image stitching. In: *IEEE International Conference on Image Processing (ICIP)*, pp. 56–60 (2014)
8. Park, J., Tai, Y.W., Sinha, S.N., Kweon, I.S.: Efficient and robust color consistency for community photo collections. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 430–438 (2016)
9. Naik, S.K., Murthy, C.A.: Hue-preserving color image enhancement without gamut problem. *IEEE Trans. Image Process. (TIP)* **12**, 1591–1598 (2003)
10. Solomon, C., Breckon, T.: *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley, Hoboken (2011)
11. Russ, J.C.: *The Image Processing Handbook*, 6th edn. CRC Press, Boca Raton (2011)
12. Sapiro, G., Caselles, V.: Histogram modification via differential equations. *J. Differ. Equ.* **135**, 238–268 (1997)
13. Palma-Amestoy, R., Provenzi, E., Bertalmio, M., Caselles, V.: A perceptually inspired variational framework for color enhancement. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* **31**, 458–474 (2009)
14. Provenzi, E.: Perceptual color correction: a variational perspective. In: Trémeau, A., Schettini, R., Tominaga, S. (eds.) *CCIW 2009. LNCS*, vol. 5646, pp. 109–119. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-03265-3_12](https://doi.org/10.1007/978-3-642-03265-3_12)
15. Nikolova, M., Steidl, G.: Fast ordering algorithm for exact histogram specification. *IEEE Trans. Image Process. (TIP)* **23**, 5274–5283 (2014)

16. Nikolova, M., Steidl, G.: Fast hue and range preserving histogram specification: theory and new algorithms for color image enhancement. *IEEE Trans. Image Process. (TIP)* **23**, 4087–4100 (2014)
17. Pierre, F., Aujol, J.-F., Bugeau, A., Ta, V.-T.: Luminance-hue specification in the RGB space. In: Aujol, J.-F., Nikolova, M., Papadakis, N. (eds.) *SSVM 2015*. LNCS, vol. 9087, pp. 413–424. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-18461-6_33](https://doi.org/10.1007/978-3-319-18461-6_33)
18. Rizzi, A., Gatta, C., Marini, D.: A new algorithm for unsupervised global and local color correction. *Pattern Recogn. Lett.* **24**, 1663–1677 (2003)
19. Yuan, L., Sun, J.: Automatic exposure correction of consumer photographs. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012*. LNCS, vol. 7575, pp. 771–785. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-33765-9_55](https://doi.org/10.1007/978-3-642-33765-9_55)
20. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Comput. Graph. Appl.* **21**, 34–41 (2001)
21. Papadakis, N., Provenzi, E., Caselles, V.: A variational model for histogram transfer of color images. *IEEE Trans. Image Process. (TIP)* **20**, 1682–1695 (2011)
22. Hristova, H., Le Meur, O., Cozot, R., Bouatouch, K.: Style-aware robust color transfer. In: *Proceedings of the workshop on Computational Aesthetics*, pp. 67–77 (2015)
23. Hwang, Y., Lee, J.Y., Kweon, I.S., Kim, S.J.: Color transfer using probabilistic moving least squares. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3342–3349 (2014)
24. Faridul, H.S., Pouli, T., Chamaret, C., Stauder, J., Reinhard, E., Kuzovkin, D., Tremeau, A.: Colour mapping: a review of recent methods, extensions and applications. *Comput. Graph. Forum* **35**, 59–88 (2016)
25. HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D.: Optimizing color consistency in photo collections. *ACM Trans. Graph. (TOG)* **32**, Article No. 38 (2013)
26. HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D.: Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph. (TOG)* **30**, Article No. 70 (2011)
27. Panetta, K., Gao, C., Agaian, S.: No reference color image contrast and quality measures. *IEEE Trans. Consum. Electron.* **59**, 643–651 (2013)
28. Bringier, B., Richard, N., Larabi, M.C., Fernandez-Maloigne, C.: No-reference perceptual quality assessment of colour image. In: *the 14th European Signal Processing Conference (EUSIPCO)*, pp. 1–5 (2006)
29. Agaian, S.S., Silver, B., Panetta, K.A.: Transform coefficient histogram-based image enhancement algorithms using contrast entropy. *IEEE Trans. Image Process. (TIP)* **16**, 741–758 (2007)
30. Agaian, S.S., Panetta, K., Grigoryan, A.M.: A new measure of image enhancement. In: *IASTED International Conference on Signal Processing & Communication*, pp. 19–22 (2000)