



Article

# Histogram-Based Color Transfer for Image Stitching <sup>†</sup>

Qi-Chong Tian and Laurent D. Cohen \*

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

\* Correspondence: cohen@ceremade.dauphine.fr; Tel.: +33-1-4405-4678

† This paper is an extended version of our paper published in the 6th International Conference on Image Processing Theory, Tools and Applications (IPTA'16), Oulu, Finland, 12–15 December 2016.

Received: 5 July 2017; Accepted: 6 September 2017; Published: 9 September 2017

**Abstract:** Color inconsistency often exists between the images to be stitched and will reduce the visual quality of the stitching results. Color transfer plays an important role in image stitching. This kind of technique can produce corrected images which are color consistent. This paper presents a color transfer approach via histogram specification and global mapping. The proposed algorithm can make images share the same color style and obtain color consistency. There are four main steps in this algorithm. Firstly, overlapping regions between a reference image and a test image are obtained. Secondly, an exact histogram specification is conducted for the overlapping region in the test image using the histogram of the overlapping region in the reference image. Thirdly, a global mapping function is obtained by minimizing color differences with an iterative method. Lastly, the global mapping function is applied to the whole test image for producing a color-corrected image. Both the synthetic dataset and real dataset are tested. The experiments demonstrate that the proposed algorithm outperforms the compared methods both quantitatively and qualitatively.

**Keywords:** color transfer; color correction; image stitching; histogram specification; global mapping curve

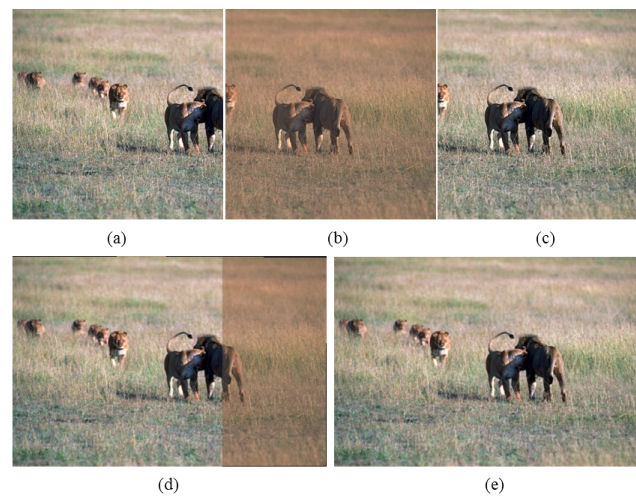
## 1. Introduction

Image stitching [1] is the technique for producing a panorama large-size image from multiple small-size images. Due to the differences in imaging devices, camera parameter settings or illumination conditions, these multiple images are usually color inconsistent. This will affect visual results of image stitching. Thus, color transfer plays an important role in image stitching. It can maintain the color consistency and make the panorama be more natural than the results without color transfer.

Color transfer is also known as color correction, color mapping or color alignment in the literature [2–7]. This kind of technique is aimed to transfer the color style of a reference image to a test image. It can make these images to be color consistent. One example is shown in Figure 1, from which we can obviously see the effectiveness of color transfer in image stitching.

Pitie et al. [8,9] proposed an automated color mapping method using color distribution transfer. There are two parts in their algorithm. The first part is to obtain a one-to-one color mapping using three-dimensional probability density function transfer, which is iterative, nonlinear and convergent. The second part is to reduce grain noise artifacts via a post-processing algorithm, which adjusts the gradient field of the corrected image to match the test image. Fecker et al. [10] proposed a color correction algorithm using cumulative histogram matching. They used basic histogram matching algorithm for the luminance component and chrominance components. Then, the first and last active bin values of cumulative histograms are modified to satisfy the monotonic constraint, which can avoid possible visual artifacts. Nikolova et al. [11,12] proposed a fast exact histogram specification algorithm, which can be applied to color transfer. This approach relies on an ordering algorithm, which is based

on a specialized variational method [13]. They used a fast fixed-point algorithm to minimize the functions and obtain color corrected images.



**Figure 1.** An example of color transfer in image stitching. (a) reference image; (b) test image; (c) color transfer for the test image using the reference color style; (d) stitching without color transfer; (e) stitching with color transfer. Image Source: courtesy of the authors and databases referred on [2,14].

Compared to the previous approaches described above, we combine the ideas of histogram specification and global mapping to produce a color transfer function, which can extend well the color mapping from the overlapping region to the entire image. The main advantage of our method is the color transfer ability for two images having small overlapping regions. The experiments also show that the proposed algorithm outperforms other methods in terms of objective evaluation and subjective evaluation.

This paper is an extended version of our previous work [15]. Compared with the conference paper [15], more related work are introduced, more comparisons and discussions are included in this paper. The rest of this paper is organized as follows. The related work is summarized in Section 2. The detailed proposed color transfer algorithm is presented in Section 3. The experiments and the result analysis are shown in Section 4. The discussion and conclusion are given in Section 5.

## 2. Related Work

Image stitching approaches can combine multiple small-size images together to produce a large-size panorama image. Generally speaking, image alignment and color transfer are the two important challenging tasks in image stitching, which has received a lot of attention recently [1,16–20]. Different kinds of image alignment methods or different color transfer algorithms can construct different approaches for image stitching. Even though color transfer method is the main topic studied in this paper, we also introduce the image alignment algorithms to make this research be comprehensive and be understood easily. A brief review of the methods for image alignment and color transfer is presented below.

### 2.1. Image Alignment

Motion models describe the mathematical relationships between the pixel coordinates in one image and the pixel coordinates in the other image. There are four main kinds of motion models in image stitching, including 2D translations, 3D translations, cylindrical and spherical coordinates, and lens distortions. For a specific application, a corresponding motion model needs to be defined first. Then, the parameters in the motion model can be estimated using corresponding algorithms. At last, the considered images can be aligned rightly to create a panorama image. We summarize two kinds of alignment algorithms, including pixel-based alignment and feature-based alignment.

### 2.1.1. Pixel-Based Alignment

The pixel-based alignment methods are to shift or warp the images relative to other images and to compare the corresponding pixels. Generally speaking, an error metric is firstly defined to compare the difference between the considered images. Then, a suitable search algorithm is applied to obtain the optimal parameters in the motion model. The detailed techniques and the comprehensive description are available in [1]. A simple description of this method is given below.

Given an image  $I_0(x_i)$ , the goal is to obtain where it is located in the other image  $I_1(x_i)$ . The simplest solution is to compute the minimum of the sum of squared difference function:

$$E(u) = \sum_i (I_1(x_i + u) - I_0(x_i))^2 = \sum_i e_i^2, \tag{1}$$

where  $u$  is the displacement vector,  $e_i = I_1(x_i + u) - I_0(x_i)$  is the residual error. To solve this minimization problem, the search algorithms will be adopted. The simplest method is the full search technique. For speeding up the computation, coarse-to-fine techniques based on image pyramids are often used in the practical applications.

### 2.1.2. Feature-Based Alignment

The feature-based alignment methods are to extract distinctive features (interesting points) from each image and to match every feature. Then, the geometric transformation between the considered images is estimated. The most popular feature extraction method is the scale-invariant feature detection [21]. The most widely used solution for feature matching is the indexing schemes based on finding nearest neighbors in high-dimension spaces. For estimating the geometric transformation, a usual method is to use least squares to minimize the sum of squared residuals by

$$E_{LS} = \sum_i \|r_i\|^2 = \sum_i \|\hat{x}'_i(x_i; p) - \hat{x}_i\|^2, \tag{2}$$

where  $\hat{x}'_i$  is the detected feature point location corresponding to point  $x_i$  in other images,  $\hat{x}_i$  is the estimated location, and  $p$  is the estimated motion parameter. Equation (2) assumes all feature points are matched with the same accuracy, which does not work well in the real application. Thus, the weighted least square is often used to obtain more robust results via

$$E_{WLS} = \sum_i \sigma_i^{-2} \|r_i\|^2, \tag{3}$$

where  $\sigma_i^2$  is a variance estimate.

## 2.2. Color Transfer

The color transfer problem is well reviewed in [2,5]. A brief introduction is summarized below.

### 2.2.1. Geometry-Based Color Transfer

Geometric-based color transfer methods compute the color mapping functions using the corresponding feature points in multiple images. Feature detection algorithms are adopted to obtain the interesting points. Scale-Invariant Feature Transform (SIFT) [21] and Speeded-Up Robust Feature (SURF) [22] are the two most widely used methods for feature detection. After obtaining the features of each image, the correspondences between the considered images are matched using the Random Sample Consensus algorithm (RANSAC), which can remove the outliers efficiently to improve the matching accuracy. Then, the correspondences are used to build a color transfer function via minimizing the color difference between the corresponding feature points. Finally, this transfer function is applied to the target image to produce the color transferred image.

### 2.2.2. Statistics-Based Color Transfer

When the feature detection and matching are not available, the geometry-based color transfer can not work. In this situation, the statistical correlation [23] between the reference image and the test image is used to create the color mapping function, which can transfer the color style of the reference image to the test image and enforce the considered images to share the same color style. Reinhard et al. [24] proposed a simple and traditional statistics-based algorithm to transfer colors between two images, which was also extended by many researchers. Papadakis et al. [25] proposed a variational model for color image histogram transfer, which used the energy functional minimization to finish the goal of transferring the image color style and maintaining the image geometry. Hristova et al. [26] presented a style-aware robust color transfer method, which was based on the style feature clustering and the local chromatic adaptation transform.

### 2.2.3. User-Guided Color Transfer

When the feature matching information and the statistical information of the considered images are both difficult to be obtained, it is essential to adopt user-guided methods to create the correspondences and use them to build the color transfer mapping function. The transfer function between images can be obtained from a set of strokes [27], which are user-defined by painting on the considered images. Then, the transfer function can be computed via different minimization approaches. The other kind of method is the color swatch based algorithm [28], which is more related to the construction of the correspondences between the considered images. The color mapping function is obtained from swatched regions in one image and can be applied to the corresponding regions in the other image.

## 3. The Proposed Approach

This paper proposes a method of color transfer in image stitching using histogram specification and global mapping. Generally speaking, there are four steps in this algorithm. Firstly, there are two given images to be stitched. The image with good visual quality is defined as the reference image, and the other is defined as the test image. Overlapping regions between these two images are obtained using a feature-based matching method. Secondly, histogram specification is conducted for the overlapping regions. Thirdly, using corresponding pixels in the overlapping region, which are original pixels and the pixels after histogram specification, the mapping function is computed with an iterative method for minimizing color differences. At last, the whole color transferred image is produced by applying the mapping function to the entire test image.

### 3.1. The Notations and the Algorithm Framework

$R$  is a reference image,

$T$  is a test image,

$R\_O$  is the overlapping region in the reference image,

$T\_O$  is the overlapping region in the test image,

$T\_O\_HS$  is the result of histogram specification for  $T\_O$ ,

$(i, j)$  is the location of pixels in images,

$k$  is the pixel values,  $k \in [0, 1, \dots, 255]$  for 8-bit images,

$\varepsilon(k) := \{(i, j) \in T\_O \mid T\_O(i, j) = k\}$ ,

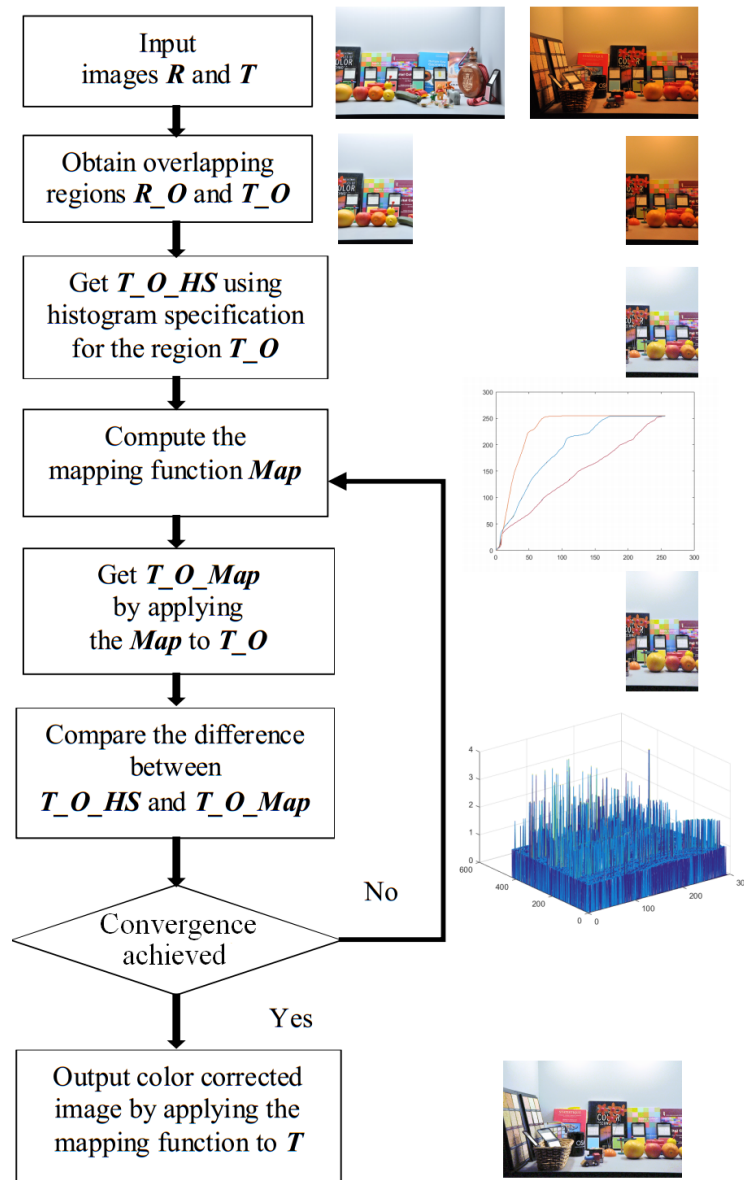
$Map$  is a color mapping function,

$T\_O\_Map$  is the result of color transfer for  $T\_O$  using the color mapping function,

$Diff$  is pixel differences between two images,

PSNR is the peak signal-to-noise ratio between two images.

The algorithm framework is described in Figure 2.



**Figure 2.** The framework of the proposed algorithm. Image Source: courtesy of the authors and databases referred on [29].

### 3.2. The Detailed Description of This Algorithm

In this section, we will describe the proposed algorithm in detail.

#### 3.2.1. Obtain Overlapping Regions between Two Images

In the application of image stitching, there are overlapping regions between input images. Due to little changes of scenes, differences of image capture angles, differences of focal lengths and other factors, the corresponding overlapping regions are not exactly pixel-to-pixel. Firstly, we find matching points between the reference image and the test image, using the scale-and-rotation-invariant feature descriptor (SURF) [22]. Then, the geometric transformation will be estimated from the corresponding points. In our implementation, the projective transformation is applied. After that, these images can be transformed and placed to the same panorama [1]. At last, we obtain overlapping regions using the image correspondence location information. This part is described in Algorithm 1.

---

**Algorithm 1** Obtain overlapping regions between two images.

---

- 1: Input two images  $R$  and  $T$ , then compute the feature point correspondences  $R_i \leftrightarrow T_i$  using SURF,  $i = 1, 2, \dots, N$ , where  $N$  is the number of feature point correspondences.
- 2: Estimate the geometric transform  $tforms$  using the correspondences, the following term is minimized:

$$\min \sum_{i=1}^N (R_i - tform(T_i))^2.$$

- 3: Warp these two images and put in the same panorama using the geometric transform  $tforms$ , define two matrixes  $M_1$  and  $M_2$  to store position information.
  - 4: Obtain overlapping regions using the image correspondence location information described in  $M_1$  and  $M_2$ .
- 

### 3.2.2. Histogram Specification for the Overlapping Region

In this step, we will make exact histogram specification for the overlapping region in the test image to match the histogram of the overlapping region in the reference image. The histogram is calculated as follows:

$$Hist(k) = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \delta[k, T(i, j)], \quad (4)$$

where

$$\delta[a, b] = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise.} \end{cases}$$

$T$  is an image,  $k$  are pixel values,  $k \in [0, 1, \dots, 255]$  for 8-bit images,  $m$  and  $n$  are the height and width of the image, and  $i$  and  $j$  are the columns and rows of pixels.

Histogram specification is also known as histogram matching, which is aimed to transform an input image to an output image fitting a specific histogram. We adopt an algorithm in [11] to perform the histogram specification in overlapping regions between the reference image and the test image. The detailed algorithm is described in Algorithm 2.

---

**Algorithm 2** Histogram specification for the overlapping region.

---

- 1: Input:  $T\_O$  is the overlapping region in the test image,  $hist$  is the histogram of  $R\_O$ ,  $u^{(0)} = T\_O$ ,  $\alpha = 0.05$ ,  $\beta = 0.1$ , iteration number  $S = 5$ ,  $c_0 = 0$ .
- 2: For  $s = 1, \dots, S$ , compute

$$u^{(s)} = T\_O - \eta^{-1}(\beta \nabla^T \eta(\nabla u^{(s-1)})),$$

- where  $\nabla$  is the gradient operator,  $\nabla^T$  is the transposition of  $\nabla$ ,  $\eta^{-1}(x) = \frac{\alpha x}{1-|x|}$ ,  $\eta(x) = \frac{x}{\alpha+|x|}$ .
- 3: Order the values in  $\Pi_N$  according to the corresponding ascending entries of  $u^{(s)}$ , where  $\Pi_N := \{1, \dots, N\}$  denote the index set of pixels in  $T\_O$ .
  - 4: For  $k = 0, 1, \dots, 255$ ,  
 set  $c_{(k+1)} = c_{(k)} + hist_{(k)}$  and  $T\_O\_HS[c_{(k)} + 1] = \dots = T\_O\_HS[c_{(k+1)}] = k$ .
-

### 3.2.3. Compute the Color Mapping Function

In this step, we will get the color mapping function from corresponding pixels in  $T\_O$  and  $T\_O\_HS$ . This operation is conducted for the three color channels, respectively.

For each color channel, a mapping function is computed as follows:

$$Map(k) = \left\lfloor \left( \left( \min_c \sum_{(i,j) \in \varepsilon(k)} (T\_O\_HS(i,j) - c)^2 \right) + 0.5 \right) \right\rfloor, \quad (5)$$

where  $k \in [0, 1, \dots, 255]$  for 8-bits images,  $\lfloor(x)\rfloor$  is the nearest integer of  $x$  towards minus infinity,  $\varepsilon(k) := \{(i, j) \in T\_O \mid T\_O(i, j) = k\}$ . The nearest integer of  $c$  is the mapping value corresponding to  $k$ . In the minimization problem of Equation (5), the value of  $c$  is usually not an integer. Thus, we use the nearest integer as the corresponding mapping value of  $k$ .

During the estimation of a color mapping function, we embed some constraint conditions like the related methods [3,30]. Firstly, the mapping function must be monotonic. Secondly, some function values may be obtained by interpolation methods, due to some pixel values  $k$  not existing in the overlapping regions. In our implementation, the simple linear interpolation is used. The detailed algorithm is described in Algorithm 3.

---

**Algorithm 3** Compute the color mapping function.

---

- 1: Input:  $T\_O$  is the overlapping region in the test image,  $T\_O\_HS$  is the result of histogram specification for  $T\_O$ . The following steps will be conducted for the three color channels, respectively.
- 2: For  $k = 0, 1, \dots, 255$ , minimize the function:

$$Map(k) = \left\lfloor \left( \left( \min_c \sum_{(i,j) \in \varepsilon(k)} (T\_O\_HS(i,j) - c)^2 \right) + 0.5 \right) \right\rfloor.$$

- 3: For some value of  $k$ , the set  $\varepsilon(k)$  is the empty set. Then, the corresponding  $k$  can not be computed in the above step and will be obtained using interpolation methods.
- 

### 3.2.4. Minimize Color Differences Using an Iterative Method

Firstly, color transfer is conducted in the overlapping region  $T\_O$  by the color mapping function obtained at the previous step. The result is denoted as  $T\_O\_Map$ . Secondly, pixel value differences  $Diff$ , and the PSNR between  $T\_O\_HS$  and  $T\_O\_Map$  is computed. Thirdly, the pixels  $(i, j)$  will be removed from  $\varepsilon(k) := \{(i, j) \in T\_O \mid T\_O(i, j) = k\}$ , when  $Diff(i, j)$  is larger than the preset threshold  $\mathbf{Thd\_Diff}$ , since this kind of pixel is considered to be outliers. Finally, a new color mapping function can be obtained by the algorithm described in Algorithm 3.

Repeat these processes until reaching the preset iteration times or PSNR increase is smaller than the preset threshold  $\mathbf{Thd\_PSNR}$ . After these iterations, the final mapping function is applied to the whole test image. Then, the corrected image shares the same color style with the reference image. In other words, the two images are color consistent, which are suitable for image stitching. The detailed algorithm is described in Algorithm 4.

---

**Algorithm 4** Minimize color differences using an iterative method.

---

- 1: Input:  $T\_O$  is the overlapping region in the test image,  $Map$  is the color mapping function obtained in **Algorithm 3**,  $\varepsilon(k) := \{(i, j) \in T\_O \mid T\_O(i, j) = k\}$ , maximal iteration number  $S$ ,  $Thd\_Diff$  is a threshold value,  $Thd\_PSNR$  is a threshold value.
- 2: Obtain  $T\_O\_Map$  by applying  $Map$  to  $T\_O$ , using

$$T\_O\_Map(i, j) = Map(T\_O(i, j)).$$

- 3: Compute pixel-to-pixel differences by

$$Diff(i, j) = |T\_O\_Map(i, j) - T\_O\_HS(i, j)|.$$

- 4: Remove pixels  $(i, j)$  from  $\varepsilon(k)$ , when  $Diff(i, j)$  is larger than the preset threshold  $Thd\_Diff$ .
  - 5: Compute the PSNR increase for  $T\_O\_Map$ .
  - 6: With the new sets  $\varepsilon(k)$ , repeat **Algorithm 3** and steps 2 to 5 in **Algorithm 4** until reaching the maximal iteration number or PSNR increase is smaller than the threshold  $Thd\_PSNR$ .
- 

## 4. Experiments

### 4.1. Test Dataset and Evaluation Metrics

Both synthetic image pairs and real image pairs are selected to compose the test dataset. Test images in this dataset are chosen from [2,3,14,29]. The synthetic data includes 40 reference/test image pairs. Each pair is from the same image, but with different color style. The image with good visual quality is assigned as a reference image, and the other is assigned as a test image. The real data includes 35 reference/test image pairs. These image pairs are taken under different capture conditions, including different exposures, different illuminations, different imaging devices or different capture time. For each pair, the image of good quality is assigned as a reference image and the other as a test image.

Anbarjafari [31] proposed an objective no-reference measure for illumination assessment. Xu and Mulligan [2] proposed an evaluation method for color correction in image stitching, which has been adopted in our evaluation. This method includes two components: color similarity between a corrected image  $G$  and a reference image  $R$ , and structure similarity between a corrected image  $G$  and a test image  $T$ .

The Color Similarity  $CS(G, R)$  is defined as  $CS(G, R) = PSNR(G\_O, R\_O)$ . PSNR is the Peak Signal-to-Noise Ratio [32] and  $G\_O, R\_O$  are the overlapped regions of  $G$  and  $R$ , respectively. The higher value of  $CS(G, R)$  indicates the more similar color style between the corrected image and the reference image. The definition of  $PSNR$  is given by

$$PSNR(A, B) = 10 \times \log_{10}\left(\frac{L^2}{MSE(A, B)}\right),$$

$$MSE(A, B) = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (A(i, j) - B(i, j))^2, \tag{6}$$

where  $A$  and  $B$  are the considered images,  $L = 255$  for 8-bit images, and  $m$  and  $n$  are the height and width of the considered images.

The structure similarity  $SSIM(G, T)$  is the Structural SIMilarity index, which is defined as a combination of luminance, contrast and structure components [33]. The higher value of  $SSIM(G, T)$



indicates the more similar structure between the corrected image and the test image. The definition of *SSIM* is described by

$$SSIM(A, B) = \frac{1}{N} \sum_{i=1}^N SSIM(a_i, b_i), \tag{7}$$

where  $N$  is the number of local windows for an image, and  $a_i, b_i$  are the image blocks at the  $i$ th local window of the image  $A$  and  $B$ , respectively. The detailed computation of  $SSIM(a_i, b_i)$  is described by

$$SSIM(a, b) = [l(a, b)]^\alpha \times [c(a, b)]^\beta \times [s(a, b)]^\gamma, \tag{8}$$

where  $l(a, b) = \frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1}$ ,  $c(a, b) = \frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2}$ ,  $s(a, b) = \frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3}$ ,  $\mu_a$  and  $\mu_b$  are the mean luminance values of the windows  $a$  and  $b$ , respectively,  $\sigma_a$  and  $\sigma_b$  are the standard variance of the windows  $a$  and  $b$ , respectively,  $\sigma_{ab}$  is the auto-covariance between the windows  $a$  and  $b$ ,  $C_1, C_2, C_3$  are small constants to avoid divide-by-zero error, and  $\alpha, \beta, \gamma$  are constants controlling the weight among the three components. The default settings recommended in [33] are:  $C_1 = (0.01L)^2, C_2 = (0.03L)^2, C_3 = \frac{C_2}{2}, L = 255, \alpha = \beta = \gamma = 1$ .

In the following parts, we compare our algorithm with the methods proposed in [9–11]. These methods transfer the color style of the whole reference image to the whole test image. The source codes of Pitie’s and Nikolova’s methods are downloaded from their homepages. The source code of Fecker’s method is obtained from [2].

#### 4.2. Experiments on Synthetic Image Pairs

Each synthetic image pair from [2,14,34,35] describes the same scene (exactly pixel-to-pixel) with different color styles. Our algorithm is applied to color correction in image stitching, so we cropped these image pairs to have various overlapping percentages, which simulates the situation in image stitching. Then, color transfer methods are applied to the corresponding image pairs that have different overlapping percentages. In the following experiments, we cropped each image pair with four different overlapping percentages (10%, 30%, 60% and 80%), respectively. Thus, we have  $40 \times 4 = 160$  synthetic pairs to make numerical experiments. As shown in the Table 1, our algorithm outperforms other methods in terms of color similarity and structure similarity.

**Table 1.** Comparison for synthetic dataset (average of 40 image pairs for each overlapping percentage). CS is the Color Similarity index, SSIM is the Structural SIMilarity index.

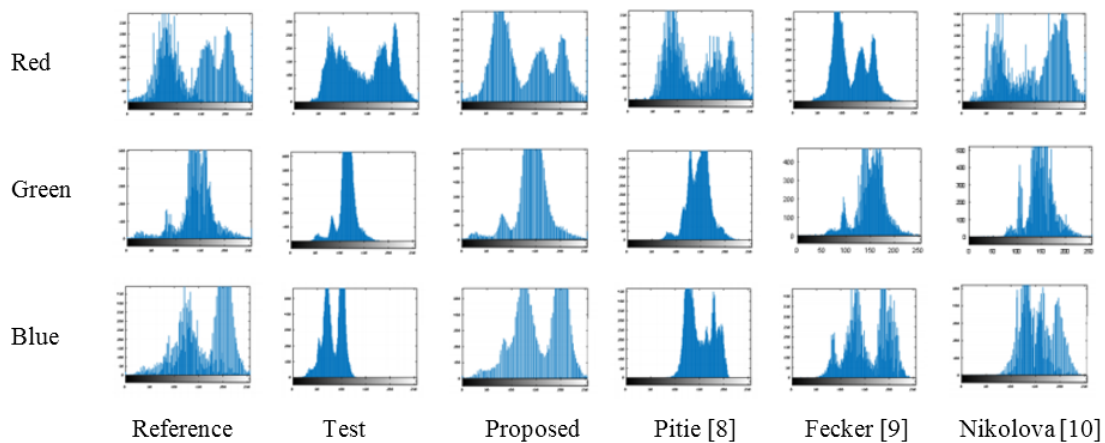
Overlapping Percentage	CS (dB)				SSIM			
	Pitie	Fecker	Nikolova	Proposed	Pitie	Fecker	Nikolova	Proposed
10%	18.21	18.34	18.39	<b>22.03</b>	0.7924	0.8033	0.8165	<b>0.8834</b>
30%	20.16	20.28	20.31	<b>24.11</b>	0.8101	0.8181	0.8299	<b>0.8867</b>
60%	21.93	21.83	22.02	<b>24.19</b>	0.8417	0.8461	0.8545	<b>0.8853</b>
80%	23.39	23.24	23.43	<b>24.31</b>	0.8662	0.8674	0.8721	<b>0.8857</b>

From the experimental results of these algorithms, we can also make a conclusion that our algorithm obtains the better visual quality of correction results even though the overlapping percentage is very small. The ability of color transfer for image pairs having narrow overlapping regions is very important in the application of image stitching. This advantage can make our color correction algorithm more suitable for image stitching. In Table 1, we can also observe that the proposed method is not significantly better than other algorithms when the overlapping percentage is very large. For example, when the overlapping percentage is 80%, the difference between the proposed method and Nikolova’s algorithm [11] is very small. Since we adopted Nikolova’s algorithm for transferring the color style in the overlapping region, the proposed method is almost the same as Nikolova’s algorithm when the overlapping percentage is close to 100%.

Some visual comparisons are shown in Figures 3–6. In Figure 3, the overlapping regions include the information describing the sky, the pyramid and the head of the camel. The red rectangles indicate the transferred color has some distance from the reference color style in the reference image. The yellow rectangle indicates the transferred color by the proposed method is almost the same as the reference color style. We can also observe easily that our algorithm transfers color information more accurately than other algorithms. For more accurate comparison, we show the histograms of the overlapping regions in Figure 4. The histograms of the overlapping regions in the reference image and in the test image are totally different. The histograms of the overlapping regions after color transfer algorithms are closer to the reference. In addition, the results by the proposed method are the closest one, which indicates the proposed method outperforms other algorithms.



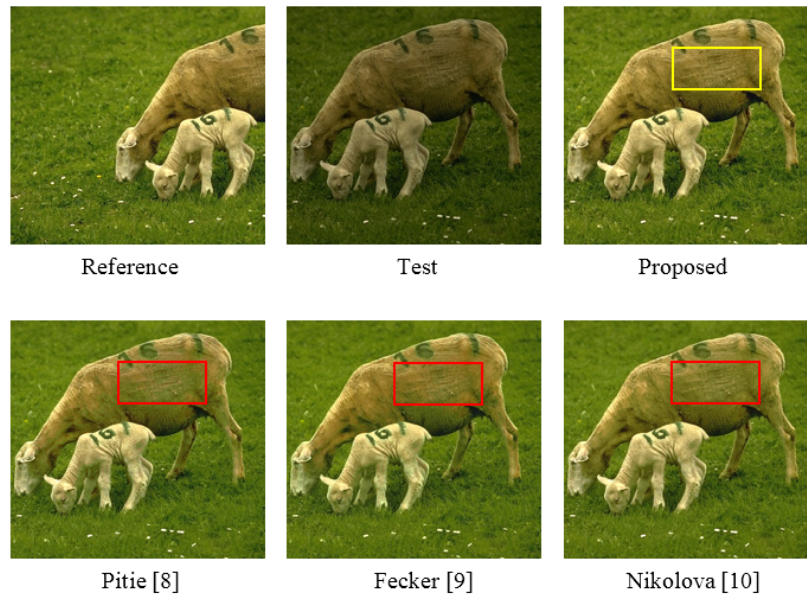
**Figure 3.** Comparison for the synthetic image pair. Image Source: courtesy of the authors and databases referred on [2,14].



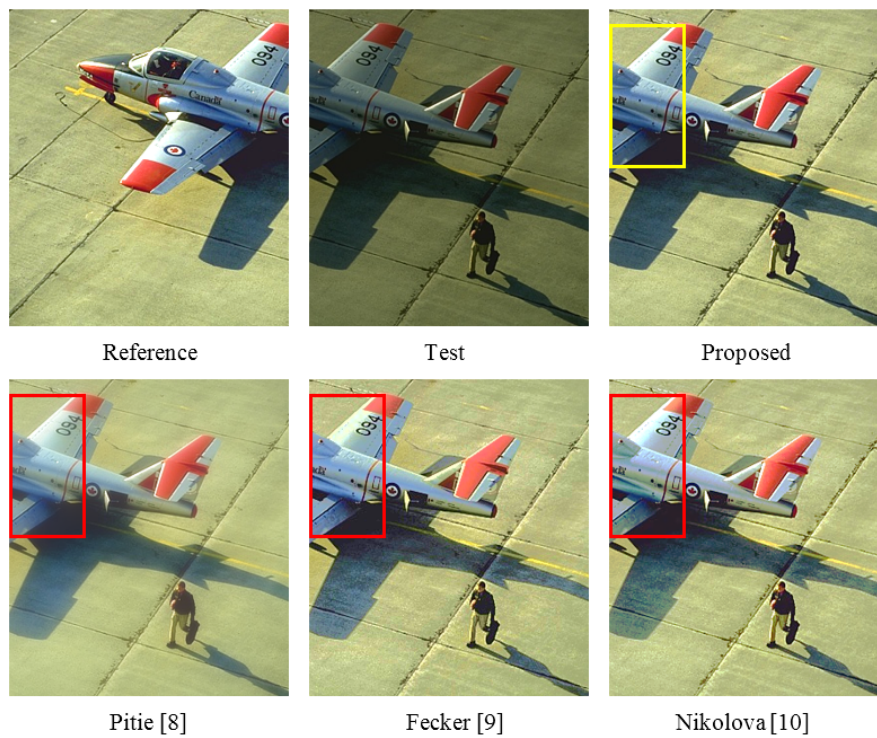
**Figure 4.** Histogram comparisons for overlapping regions in Figure 3. The first column shows the histograms (three color channels, respectively) of overlapping regions in the reference image, the second column shows the corresponding histograms in the test image, the third column shows the corresponding histograms of overlapping regions after the proposed method, the fourth column shows Pitie’s result, the fifth column shows Fecker’s result, and the last column shows Nikolova’s result.

In Figure 5, the red rectangles show disadvantages of other algorithms, which have transferred the green color to the body of the sheep. The yellow rectangle indicates the advantage of our algorithm,

which has transferred the right color to the sheep body. In Figure 6, the rectangles describe the airplane body that exists in the overlapping region. The red rectangles show disadvantages of other algorithms that transferred the inconsistent color to the airplane body. The yellow rectangle indicates that the proposed method transfers the consistent color to the airplane body.



**Figure 5.** Comparison for the synthetic image pair. Image Source: courtesy of the authors and databases referred on [2,14].

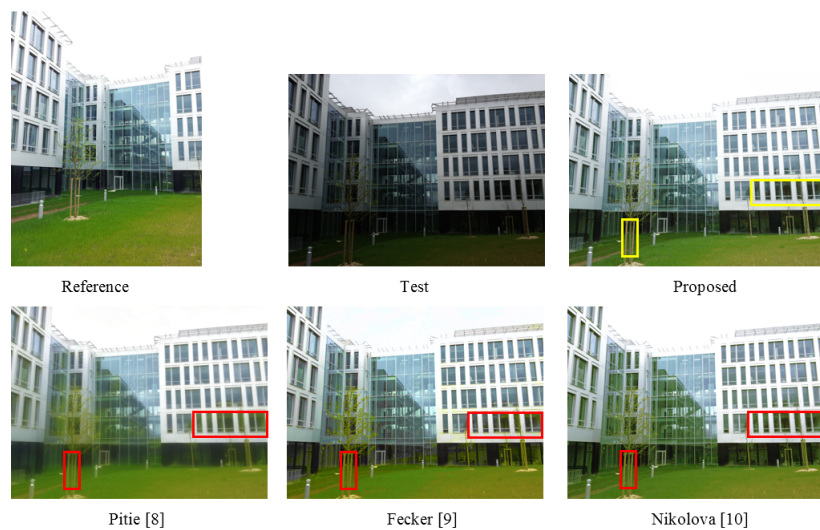


**Figure 6.** Comparison for the synthetic image pair. Image Source: courtesy of the authors and databases referred on [2,14].

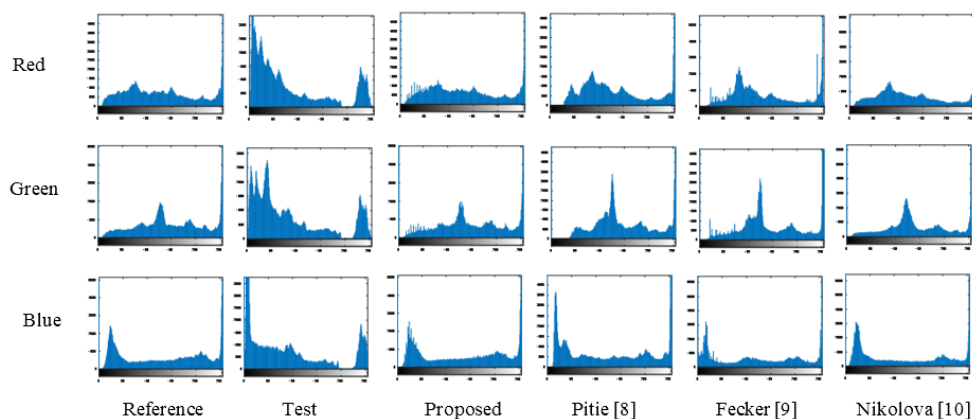
### 4.3. Experiments on Real Image Pairs

In the experiments above, we make the comparisons using synthetic image pairs, which have exactly the same overlapping regions. However, overlapping regions are not usually exactly the same (not pixel-to-pixel) in the real application of image stitching. Thus, we make some experiments for real image pairs.

Objective comparisons are given in Table 2, which indicates that our algorithm outperforms other methods in terms of color similarity and structure similarity. Subjective visual comparisons are also presented in Figures 7–10. In Figure 7, the red rectangles show disadvantages of other algorithms, which have transferred the green color to the tree body and the windows. The yellow rectangles indicate the advantage of our algorithm, which transfers the right color to the mentioned regions. The histogram comparisons for the overlapping regions are shown in Figure 8, which indicates the proposed method outperforms other algorithms. More results and the comparisons are given in Figures 9 and 10.



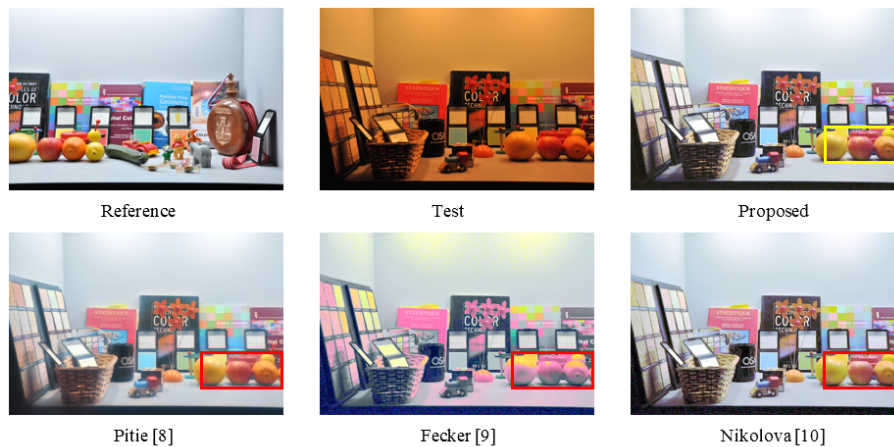
**Figure 7.** Comparison for the real image pair. Image Source: courtesy of the authors and databases referred on [29].



**Figure 8.** Histogram comparisons for overlapping regions in Figure 7. The first column shows the histograms of overlapping regions in the reference image, the second column shows the corresponding histograms in the test image, the third column shows the corresponding histograms of overlapping regions after the proposed method, the fourth column shows Pitie’s result, the fifth column shows Fecker’s result, and the last column shows Nikolova’s result.

**Table 2.** Comparison for real image pairs (Average of 35 pairs).

	<i>Pitie</i>	<i>Fecker</i>	<i>Nikolova</i>	<i>Proposed</i>
<b>CS(dB)</b>	18.98	19.04	19.12	<b>21.19</b>
<b>SSIM</b>	0.8162	0.8334	0.8255	<b>0.8531</b>



**Figure 9.** Comparison for the real image pair. Image Source: courtesy of the authors and databases referred on [29].



**Figure 10.** Comparison for the real image pair. Image Source: courtesy of the authors and databases referred on [3].

### 5. Discussion

In this paper, we have proposed an efficient color transfer method for image stitching, which combines the ideas of histogram specification and global mapping. The main contribution of the proposed method is using original pixels and the corresponding pixels after histogram specification to compute a global mapping function with an iteration method, which can effectively minimize color differences between a reference image and a test image. The color mapping function can spread well the color style from the overlapping region to the whole image. The experiments also demonstrate the advantages of our algorithm in terms of objective evaluation and subjective evaluation.

As our work relies on the exact histogram specification, bad results of histogram specification will decrease the visual quality of our results. Even though the problem of histogram specification has received considerable attention and has been well studied during recent years, some future work can be conducted to improve the results of this kind of algorithm.

In the detailed description of the proposed algorithm, we have shown that our method is building the color mapping functions using the global information and without using the local neighbor information. In future work, we will consider the information of local patches to construct the color mapping functions, which may be more accurate to transfer colors. Another problem is that the mapping function is computed for each color channel. This simple processing does not consider the relation of the three color channels, and this may produce some color artifacts. In our future work, we try to obtain the color mapping function considering the relation of the three color channels. The minimization is completed with an iteration framework, and the termination conditions include computing PSNR. These operations need high computation, so a fast minimization method will also be considered in the future work.

**Acknowledgments:** We would like to sincerely thank Hasan Sheikh Faridul, Youngbae Hwang, and Wei Xu for sharing the test images and permitting us to use the images in this paper. We greatly thank Charless Fowlkes for sharing the BSDS300 dataset for research purposes.

**Author Contributions:** Qi-Chong Tian designed the algorithm presented in this article, conducted the numerical experiments, and wrote the paper. Laurent D. Cohen proposed the idea of this research, analyzed the results, and revised the whole article. Qi-Chong Tian is a Ph.D. student supervised by Laurent D. Cohen.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Szeliski, R. Image Alignment and Stitching: A Tutorial. *Found. Trends Comput. Graph. Vis.* **2006**, *2*, 1–104.
2. Xu, W.; Mulligan, J. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), San Francisco, CA, USA, 13–18 June 2010; pp. 263–270.
3. Hwang, Y.; Lee, J.Y.; Kweon, I.S.; Kim, S.J. Color transfer using probabilistic moving least squares. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14), Columbus, OH, USA, 23–28 June 2014; pp. 3342–3349.
4. Faridul, H.; Stauder, J.; Kervic, J.; Trémeau, A. Approximate cross channel color mapping from sparse color correspondences. In Proceedings of the IEEE International Conference on Computer Vision (ICCV'13)—Workshop in Color and Photometry in Computer Vision (CPCV'13), Sydney, Australia, 8 December 2013; pp. 860–867.
5. 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* **2016**, *35*, 59–88.
6. Fitschen, J.H.; Nikolova, M.; Pierre, F.; Steidl, G. A variational model for color assignment. In Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision (SSVM'15), Lege Cap Ferret, France, 31 May–4 June 2015; Volume LNCS 9087, pp. 437–448.
7. Moulon, P.; Duisit, B.; Monasse, P. Global multiple-view color consistency. In Proceedings of the European Conference on Visual Media Production (CVMP'13), London, UK, 6–7 November 2013.
8. Pitie, F.; Kokaram, A.C.; Dahyot, R.  $N$ -dimensional probability density function transfer and its application to color transfer. In Proceedings of the IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005; Volume 2, pp. 1434–1439.
9. Pitié, F.; Kokaram, A.C.; Dahyot, R. Automated colour grading using colour distribution transfer. *Comput. Vis. Image Underst.* **2007**, *107*, 123–137.
10. Fecker, U.; Barkowsky, M.; Kaup, A. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1258–1267.
11. Nikolova, M.; Steidl, G. Fast ordering algorithm for exact histogram specification. *IEEE Trans. Image Process.* **2014**, *23*, 5274–5283.
12. Nikolova, M.; Steidl, G. Fast hue and range preserving histogram specification: Theory and new algorithms for color image enhancement. *IEEE Trans. Image Process.* **2014**, *23*, 4087–4100.
13. Nikolova, M.; Wen, Y.W.; Chan, R. Exact histogram specification for digital images using a variational approach. *J. Math. Imaging Vis.* **2013**, *46*, 309–325.

14. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV'01), Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 416–423.
15. Tian, Q.C.; Cohen, L.D. Color correction in image stitching using histogram specification and global mapping. In Proceedings of the The 6th International Conference on Image Processing Theory Tools and Applications (IPTA'16), Oulu, Finland, 12–15 December 2016; pp. 1–6.
16. Brown, M.; Lowe, D.G. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.* **2007**, *74*, 59–73.
17. Xiong, Y.; Pulli, K. Fast panorama stitching for high-quality panoramic images on mobile phones. *IEEE Trans. Consum. Electron.* **2010**, *56*, doi:10.1109/TCE.2010.5505931.
18. Wang, W.; Ng, M.K. A variational method for multiple-image blending. *IEEE Trans. Image Process.* **2012**, *21*, 1809–1822.
19. Shan, Q.; Curless, B.; Furukawa, Y.; Hernandez, C.; Seitz, S.M. Photo Uncrop. In Proceedings of the 13th European Conference on Computer Vision (ECCV'14), Zurich, Switzerland, 6–12 September 2014; pp. 16–31.
20. Lin, C.C.; Pankanti, S.U.; Natesan Ramamurthy, K.; Aravkin, A.Y. Adaptive as-natural-as-possible image stitching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15), Boston, MA, USA, 7–12 June 2015; pp. 1155–1163.
21. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
22. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359.
23. Provenzi, E. *Variational Models for Color Image Processing in the RGB Space Inspired by Human Vision*; Habilitation à Diriger des Recherches; ED 386: École doctorale de sciences mathématiques de Paris centre, UPMC, France, 2016.
24. Reinhard, E.; Adhikhmin, M.; Gooch, B.; Shirley, P. Color transfer between images. *IEEE Comput. Graph. Appl.* **2001**, *21*, 34–41.
25. Papadakis, N.; Provenzi, E.; Caselles, V. A variational model for histogram transfer of color images. *IEEE Trans. Image Process.* **2011**, *20*, 1682–1695.
26. Hristova, H.; Le Meur, O.; Cozot, R.; Bouatouch, K. Style-aware robust color transfer. In Proceedings of the workshop on Computational Aesthetics. Eurographics Association, Istanbul, Turkey, 20–22 June 2015; pp. 67–77.
27. Wen, C.L.; Hsieh, C.H.; Chen, B.Y.; Ouhyoung, M. *Example-Based Multiple Local Color Transfer by Strokes*; Computer Graphics Forum; Wiley Online Library: Hoboken, NJ, USA, 2008; Volume 27, pp. 1765–1772.
28. Welsh, T.; Ashikhmin, M.; Mueller, K. *Transferring Color to Greyscale Images*; ACM Transactions on Graphics (TOG); ACM: New York, NY, USA, 2002; Voume 21, pp. 277–280.
29. Faridul, H.S.; Stauder, J.; Trémeau, A. Illumination and device invariant image stitching. In Proceedings of the IEEE International Conference on Image Processing (ICIP'14), Paris, France, 27–30 October 2014; pp. 56–60.
30. Tai, Y.W.; Jia, J.; Tang, C.K. Local color transfer via probabilistic segmentation by expectation-maximization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), 20–25 June 2005; Volume 1, pp. 747–754.
31. Anbarjafari, G. An Objective No-Reference Measure of Illumination Assessment. *Meas. Sci. Rev.* **2015**, *15*, 319–322.
32. Maitre, H. *From Photon to Pixel: The Digital Camera Handbook*; Wiley Online Library: Hoboken, NJ, USA, 2015.
33. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612.
34. Color Correction Images. Available online: [https://www.researchgate.net/publication/282652076\\_color\\_correction\\_images](https://www.researchgate.net/publication/282652076_color_correction_images) (accessed on 2 September 2017).
35. The Berkeley Segmentation Dataset and Benchmark. Available online: <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/> (accessed on 2 September 2017).

