Computers & Graphics 35 (2011) 67–80

Contents lists available at ScienceDirect

Computers & Graphics

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

Technical Section

Progressive color transfer for images of arbitrary dynamic range

Tania Pouli *, Erik Reinhard

University of Bristol, United Kingdom

ARTICLE INFO

Available online 11 November 2010

Keywords:
Color transfer
Histogram matching
Tone reproduction
High dynamic range

ABSTRACT

Image manipulation takes many forms. A powerful approach involves image adjustment by example. To make color edits more intuitive, the intelligent transfer of a user-specified target image's color palette can achieve a multitude of creative effects, provided the user is supplied with a small set of straightforward parameters. We present a novel histogram reshaping technique which allows significantly better control than previous methods and transfers the color palette between images of arbitrary dynamic range. We achieve this by manipulating histograms at different scales, which allows coarse and fine features to be considered separately. We compare our approach to a number of existing color transfer and tonemapping techniques and demonstrate its performance for a wide range of images.

1. Introduction

Images can convey information not only through the depicted objects but also through the particular mood, color scheme and composition of the scene. Artists can manipulate the color palette manually to change the appearance of an image and achieve specific effects but that can be a time-consuming process, requiring advanced image manipulation skills. To that end, several color transfer techniques have been proposed that uses the color palette of a second image as a target and achieve similar results with minimal user input and skill necessary. Matching the color distribution of one image to another is typically achieved by transferring some characteristics between the two images ranging from simple statistical properties to more complex distribution transfers, generally focusing on preserving certain visual qualities of the image.

A limitation of existing color transfer techniques is the lack of control over how much the input image should be matched to the target. As color transfer is particularly useful for artistic purposes, intuitive and simple control over the result is a desirable feature. This would also make the selection of the target image less critical to achieve a plausible result. As a consequence, a wider range of target images could be used, opening up the possibility of using the selection of target images as a part of the creative process.

Histograms are used extensively in imaging applications as they provide a compact space in which images can be manipulated. Images of typical scenes (natural or otherwise) contain a lot of redundant information as most pixels are similar in color to their neighbors. These similarly colored image portions tend to correspond to persistent peaks in the histogram of the image, while smaller within-region variations create the higher frequency details. We therefore hypothesize that histograms can be manipulated at different scales, allowing different portions of the image to be affected without requiring manual selection or segmentation.

With this motivation, we propose a color transfer technique that can progressively reshape the histogram of a given image to match it to the histogram of another. Our approach relies on the novel idea of a scale-space manipulation of the histograms, which allows us to match features at coarser or finer scales. This is the key for achieving a range of appearances: we allow the user to select how well the color palette of the input image should be matched to that of the target. At a minimum, the result will maintain the original appearance of the source image, while at a maximum the histogram of the input will fully match that of the target. With our scale-space approach a partial match can be achieved by only reshaping the histogram according to features in coarse scales, while a full match considers finer scales too, allowing more detail to be captured (see Fig. 1).

Additionally, our approach allows colors to be transferred between images of varying dynamic ranges. When a high dynamic range (HDR) source and a low dynamic range (LDR) target are used, the input HDR image is matched to the target LDR both in color and in dynamic range. As such, the proposed technique is suitable for creatively tonemapping HDR images using a target to specify the desired appearance of the result.

We review the relevant literature in Section 2 and present our algorithm in Section 3, while Section 4 proposes two region selection mechanisms that can be used with our technique. Section 5 discusses tone reproduction with a reference using the proposed method. Lastly, a wide range of examples is shown and compared with a representative set of existing techniques in Section 6. The paper ends with a brief summary in Section 7.

* Corresponding author.
E-mail addresses: pouli@cs.bris.ac.uk (T. Pouli), reinhard@cs.bris.ac.uk (E. Reinhard).

0097-8493/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2010.11.003

Fig. 1. An example of a progressive color transfer result produced by our algorithm. (a) Source, (b) target (c) our result, partial match, (d) our result full match.

2. Related work

2.1. Color transfer

In its simplest form, the method proposed by Reinhard et al. [1] shifts and scales the pixel values of the source image to match the mean and standard deviation from the target. This is done in the $l\alpha\beta$ opponent color space, which is on average decorrelated [2]. This allows the transfer to take place independently in each channel, turning a potentially complex 3D problem into three much simpler 1D problems. Although this technique can be successful for a large range of images, the quality of the results largely depend on the composition of the source and target images.

Other global approaches transfer higher level statistical properties. Histogram matching can be used to transfer the distributions of images in a variety of color spaces. Neumann and Neumann [3] use 3D histogram matching in the HSL color space to achieve an exact match of the gamut of the target image. Histogram matching in the $l\alpha\beta$ color space is used by Xiao and Ma [4] with the addition of a post-processing step that uses optimization to preserve the gradients of the source image. To deal with larger differences in image composition, Pitié et al. [5] propose a method to transfer an N-dimensional probability distribution function to another. They use an iterative, non-linear technique that estimates the solution using 1D marginal distributions. This technique is very successful in terms of matching the color palette as it takes into account the correlations between channels, but tends to produce significant spatial artifacts. These can be removed by a somewhat involved post-process, which matches the gradient field of the output image to the input image [6].

Most color transfer techniques transfer properties in an appropriate color space that de-correlates the image data. Although color spaces such as $l\alpha\beta$ can achieve that for most images, counter-examples exist where a different set of axes would be more appropriate, i.e. better decorrelated. Using principal component analysis (PCA) Abadpour and Kasaei [7,8] compute a decorrelated color space suitable for the particular input images. They also propose a unified framework for colorizing grayscale images from colored ones. Using independent component analysis (ICA), Grundland and Dodgson [9] compute a decorrelated and independent color space that is based on the perceptually uniform CIELab color space and use an approximate histogram matching to transfer the colors between images. Similarly, in the work by Xiao and Ma [10], the image data of the source and target is decomposed into its principal components and the source pixels are transformed appropriately to match the target.

A potential source of problems with color transfer approaches in general is that if the contents of the target image are vastly different to the source, the results can look unnatural. If for instance the colors of a seascape are successfully transferred to a forest landscape, blue foliage will inevitably appear. To alleviate this effect, Chang et al. [11] propose a perception-based scheme whereby colors are classified into categories derived through a psychophysical color-naming study. The color transfer then adheres to this classification by restricting resulting colors within their original categories in order to create a natural-looking image.

Color transfer has also found applications in image and video colorization. Using luminance and texture information, Welsh et al. [12] transfer the palette of a color image to a grayscale one. Texture information is also used in Ji et al. [13] to aid in the colorization of grayscale images. Infrared video colorization has been demonstrated in Yan et al. [14] where a reference color image is used to color a monochromatic frame sequence.

Colorization or recolorization of images can also be achieved using stroke based input to define the target colors. Levin et al. [15] color grayscale images by applying simple strokes in regions of the image. The color of the strokes is then propagated to the remainder of the region using optimization-based techniques. The approach by Wen et al. [16] on the other hand, uses strokes in both the source and target image to define corresponding regions in the images rather than directly specify a color palette. Partial recolorization can also be achieved by defining a region to be altered using a simple rectangular selection that is then propagated through a color influence map [17]. More recently, An and Pellacini proposed a stroke based approach that uses pairs of strokes to define region correspondences between images [18]. Colors are transferred for each stroke pair using a nonlinear constrained parametric model that achieves a high degree of matching while minimizing artifacts. Our technique optionally employs a simpler region selection mechanism using masks to define which regions should be used in the transfer, discussed in Section 4.

A technique relatively close to ours is the one proposed by Senanayake and Alexander [19] which aims to eliminate color variations in images of similar scenes that may be due to varying illumination or viewpoint. The source and target histograms are aligned based on corresponding features that are detected as persistent peaks through scale space using a polynomial mapping. Although their technique works well when the source and target images are similar, it is not appropriate for cases where the histograms are significantly different as no corresponding features exist. In contrast, our proposed algorithm aims to transfer properties between potentially very different images. By expanding on the notion of histograms in a scale space, we are able to reshape the source histogram in order to create peaks that match the target.

2.2. Tone reproduction

High dynamic range imaging and related topics have garnered much interest in recent years, with the film and games industries counted as early adopters. Tone reproduction takes a central place in this field, given the need to reduce the dynamic range of images for display on specific devices [20]. Dynamic range reduction is often inspired by aspects of human vision, aiming to reproduce visual attributes such as contrast, brightness, or more generally appearance.

Most simple operators use a specific functional form, such as logarithms [21], histograms [22] or sigmoids [23]. The tone curves can be adapted to specific images by adjusting user parameters, giving limited but often sufficient control. Histogram adjustment is perhaps most adaptive to the input material, as it is a form of histogram equalization, albeit with a reshaping step that avoids local expansion of contrasts [22]. Spatially varying techniques locally adapt to the image, and tend to afford better compression, albeit at a higher computational cost [24–26].

None of the existing methods, however, were designed with creative control over the final result in mind. In this paper, we explore a solution whereby a high dynamic range image is tonemapped by matching it against a conventional image with an appropriate dynamic range and color composition. The creative aspect of this technique lies in the selection of the target image, rather than in the choice of user parameters. Alternatively, it would be possible to tonemap an HDR image first and then apply a color transfer algorithm. We note, however, that although this is generally a viable approach, with our technique it is not necessary to take such an indirect route.

2.3. Manipulating other image modalities

In this work, we focus on transferring the color properties from one image to another. Color however is not the only image property contributing to its look and feel. Other modalities, such as texture, contrast or overall tone can be transferred between pairs of images [27] or manipulated within a single image [28,29] to achieve a variety of effects. Alternatively, if additional input is available, demonstrating a particular relation between two images, this relation can be replicated on novel images using *image analogies* [30].

Bae et al. [27] transfer the tonal balance and texture information between two images. By decomposing the image into a base and a detail (texture) layer, they separately modulate both using a given target to match a variety of photographic looks, including pairs of LDR and HDR images. Unlike our technique though, their approach only operates on the luminance channel of the images.

Creative adjustment of tonal values (and consequently creative tone reproduction) can also be achieved by manipulating the relevant properties of an image without using a reference. Lischinski et al. [29] propose a method where the user can define regions of interest in the image using simple strokes. The selected regions can then be adjusted to achieve a variety of results. To achieve a finer level of control for tonal adjustment, Farbman et al. [28] propose a multi-scale decomposition of the image from the coarsest to the finest level of detail that can then be manipulated separately.

These techniques allow a variety of creative effects and styles to be achieved with simple user interaction. However, in contrast to our approach color is not explicitly handled in any of these cases. Only the luminance channel of color images is matched to the target or edited by the provided tools, while the chromatic channels of the source are maintained.

3. Algorithm

The goal of our algorithm is to robustly transfer the color palette between two images while allowing the user to control the amount of matching in a simple way. We achieve this by means of a novel progressive histogram reshaping approach, outlined in Fig. 2, that can produce results that span the range of appearances between the source and the target images.

Persistent maxima in the image histogram correspond to important color clusters in the image, as shown in Fig. 3. Color histograms can have high frequency features too but these tend to correspond to smaller color variations, both in color range and spatial location. By employing a scale space approach, smaller features in the histogram can be discounted to achieve a partial match or considered in the matching process to fully transfer the color palette of the target. By reshaping the source histogram only with respect to coarse features of the target, the general color style can be transferred. But it is the finer features of the histogram that ultimately contribute to the appearance and 'look and feel' of the image, and these require the transfer of finer details.

These notions can be implemented as follows. First, the input images are converted to the CIELab color space, which is a color opponent space, and thereby offers sufficiently decorrelated axes for most inputs. A D65 white point is assumed for both the conversions to CIELab and back to RGB. In this space we compute and manipulate the histograms separately for each channel. For simplicity, we will describe the steps of our algorithm for a single channel only, although the same steps should be followed for all three channels. To this end, references to an image I should be applied to each of the channels of I in turn.

The shape of a histogram can be characterized by a series of local maxima and minima. These features occur at different scales which allows them to be classified in terms of their importance. To achieve a progressive match between the two images, we take advantage of this property of histograms. To achieve a softer, partial match we can

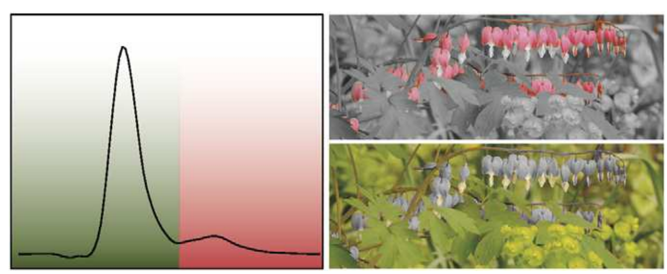

Fig. 3. The histogram shown is computed from the red–green opponent channel of the image that was first converted to the CIELab color space. At this particular scale, only two major maxima are visible. The first corresponds to the green portions in the image (bottom right) and the second to the red portions of the flowers (top right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

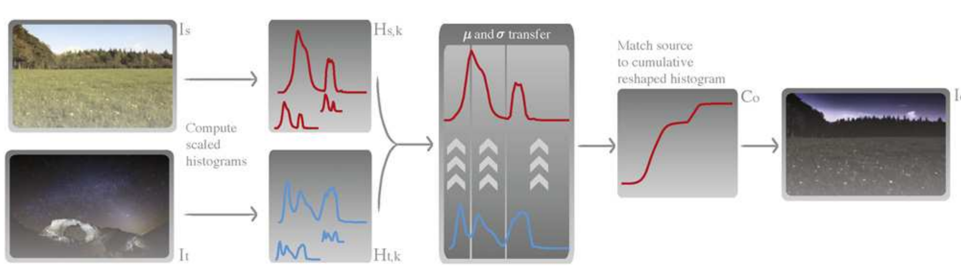

Fig. 2. Our algorithm transfers the color palette between two images by first computing their respective histograms in several scales, detecting minima in each histogram and reshaping the source histogram using the mean and standard deviation of the target for each region between the detected minima. Once the source is reshaped appropriately, the resulting image is produced by matching the source image to the cumulative reshaped histogram.

consider only coarse features while for a more faithful match, we can look at finer details of the histograms in question.

To this end, we form a scale pyramid for both the source and the target histograms, where each level of the pyramid is progressively filtered to detect persistent features (peaks) of different sizes. These features define regions to be manipulated separately in order to reshape the source to match the target histogram. We view each of these regions as locally having a Gaussian distribution and use the mean and standard deviation of the counts of bins contained in each region to transfer the palette of one image to another.

By selecting the number of scales that participate in the reshaping, we afford control over the degree of color transfer. Importantly, we select only the coarsest scales, so that the high frequency content of the source histogram is preserved in the output, whereas the lower frequencies are transferred from the target. If we select all scales, the method approximates histogram matching. However, with fewer scales considered, partial transfer results can be achieved which is one of the main contributions of our technique. An overview of our approach is given in pseudocode in Fig. 8.

3.1. Background

Before we describe our approach in more detail, we provide an overview of important concepts and tools that are instrumental to our algorithm.

Histograms: For a given image I, we define its histogram H with B bins of width V as follows:

$$H = \{(h(1),v(1)),\ldots,(h(B),v(B))\} \tag{1}$$

$$B = \left\lceil \frac{\max(I)-\min(I)}{V} \right\rceil \tag{2}$$

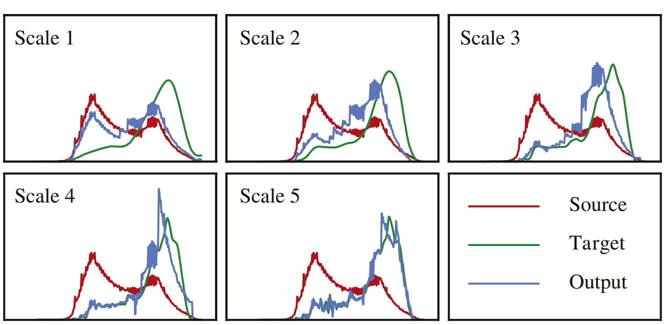

Fig. 4. Histograms for a series of consecutive scales are shown. Scale 1 is the coarsest while scale 5 is the finest. Each iteration brings the source histogram closer to the target, allowing for partial matching.

$$h(i) = \sum_{p=1}^{N} P(I(p),i), \quad i \in [1,B] \tag{3}$$

$$v(i) = \min(I)+(i-1)V \tag{4}$$

$$P(I(p),i) = \begin{cases} 1 & i = \left\lfloor \frac{I(p)-\min(I)}{V} + 1 \right\rfloor \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Fig. 6. The input images (source—left, target—right) are used with their corresponding mattes to produce the result shown at the bottom. Here, the background of target image contains some unwanted red patches. Using a mask that only includes the flower, these regions do not contribute to the color transfer. Additionally, the yellows of the target are only transferred to the flower in the source. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 5. A color to grayscale example. The palette of this distinctive Ansel Adams image is transferred to the source image with various options for the level of matching (s) and the transfer weights (w_t). As can be seen, the resulting images gradually take on the appearance of the target. (The Canyon de Chelly photograph is one of the Ansel Adams photographs available from the National Archives and Records Administration.) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

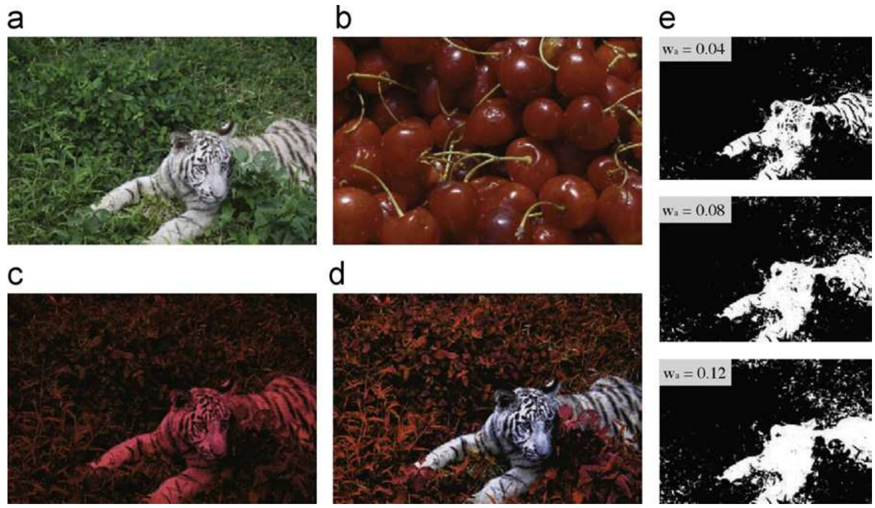

Fig. 7. The lower left image (c) is created without histogram anchoring. The tiger has acquired an unnatural red tint. The lower right image (d) demonstrates the effectiveness of histogram anchoring. Here the leaves have been correctly matched to the target but the tiger has remained white. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

```
function ReshapeHistogram(Is, It, perc)

  Convert Is, It to CIELab color space
  Compute Smax                                                            Eq.7

  for each channel ic
    Compute histograms Hs, Ht from Is(ic), It(ic)                         Eq.1–5
    for each level k in perc/Smax

      Hs,k ← Down- and up-sample Hs
      Ht,k ← Down- and up-sample Ht

      Rmin,t ← FindPeaks(Ht,k)                                            Eq.9
      for each min-bound region m in Rmin,t
        Hs,k'(Rmin,t(m):Rmin,t(m+1)) ← RegionTransfer(Hs,k(Rmin,t(m):Rmin,t(m+1)),   Eq.10–12
                                                      Ht,k(Rmin,t(m):Rmin,t(m+1)),
                                                      k/Smax)
      end

      Rmin,s ← FindPeaks(Hs,k')                                           Eq.9
      for each min bound region in Rmin,s
        Ho,k(Rmin,t(m):Rmin,t(m+1)) ← RegionTransfer(Hs,k(Rmin,s(m):Rmin,s(m+1)),    Eq.10–12
                                                     Ht,k(Rmin,s(m):Rmin,s(m+1)),
                                                     k/Smax)

      end
    end

    Io(ic) <- HistMatch (Is(ic), Ho)                                      Eq.13–15
  end

  Convert Io back to RGB for display
  return Io
end

function FindPeaks(H)                                                     Eq.9
  H(H >= 0) = 1
  H(H < 0) = -1
  H2 ← ∇H                                                                 Eq.8
  return Rmin ← find(H2 < 0)
end

function RegionTransfer(Hs,Ht,wt)
  ws ← 1-wt
  Ho ← (Hs-ws·mean(Hs))(wt·std(Ht)/ws·std(Hs)) + wt·mean(Ht)             Eq.12
  return Ho
end
```

Fig. 8. Pseudocode describing the main steps of the core part of our algorithm.

where H is the set of all pairs $(h(i),v(i))$ for all $i \in [1,B]$ corresponding to the number of elements and value of the ith bin of the histogram. $I(p)$ is the value of the pth pixel of image I which contains a total of N pixels and $P(I(p),i)$ represents the probability of a pixel $I(p)$ belonging to a bin i.

Bilateral filtering: The bilateral filter, first proposed by Tomasi and Manduchi [31] smooths regions in the image while respecting strong edges. It is generally defined as follows:

$$I_{\text{bilat}}(p) = \frac{\sum_{q \in N} f(q-p)g(I(q)-I(p))I(q)}{\sum_{q \in N} f(q-p)g(I(q)-I(p))} \tag{6}$$

where $I_{\text{bilat}}(p)$ is the output of the bilateral filter for the pth pixel of image I and f, g are Gaussians operating on pixel distances and intensities respectively. To improve the efficiency of this step, we employ the bilateral grid as presented in Chen et al. [32].

For the remainder of the paper, the subscripts s, t and o will be used to refer to the source, target and output respectively. To ensure that the size of the image does not influence the result we normalize the bin counts h_s, h_t according to the number of pixels in each image and use the same number of bins for both images.

3.2. Progressive histogram reshaping

Our approach transfers the color palette between two images of arbitrary dynamic ranges and allows for partial matches by reshaping the histogram of the image so that features of different scales can be manipulated separately.

First, we compute each scale for the target histogram H_t by downsampling the original histogram by a factor dependent on the scale that is currently computed and the maximum number of scales, $S_{\max}$. This process removes high frequency details of the histogram but preserves prominent features. Subsequently, we upsample the now smoothed histogram back to its original size to simplify further computations. Note that bicubic interpolation is used in the downsampling step, while a nearest neighbor scheme is used when upsampling back to the original histogram size as in this case no further smoothing is required. We have found, however, that different interpolation algorithms do not lead to substantially different results.

We compute $S_{\max}$ as follows:

$$S_{\max} = \left\lfloor \log_2\left(\frac{B}{B_{\min}}\right) \right\rfloor \tag{7}$$

where B is the number of bins and $B_{\min}$ is the minimum allowed histogram size. A value of 10 was used for $B_{\min}$ throughout our experiments, as in the majority of cases this means that the coarsest scale is compressed to a single peak. For each scale k, H_t is then subsampled so that it has $B_k = B\ 2^{k-S_{\max}}$ bins, where $k \in [1,S_{\max}]$. We use $k=1$ to denote the coarsest scale possible for a given histogram and $k=S_{\max}$ the finest scale. We compute the different scales of the histograms based on the bin counts only. The histogram for a given scale k is then $h_{t,k}$ (or $h_{s,k}$) and is given by downsampling by a factor of $2^{k-1+S_{\max}}$ and then upsampling back to the original size.

Next, features present in each scale of the histogram are detected. An appropriate way to achieve this is to locate zero-crossings in the first-order derivatives of the histogram. First order derivatives are computed using forward differencing:

$$\nabla h_{t,k} = h_{t,k}(i) - h_{t,k}(i+1), \quad i \in [1,B_{k-1}] \tag{8}$$

Zero crossings can be classified as minima or maxima based on the corresponding values of their second-order derivative. A first derivative of 0 corresponds to a minimum if the second derivative at the same point is positive, a maximum if it is negative and an inflection point if the second derivative is also 0. The target histogram for that scale can then be divided into a set of regions using the detected minima as follows:

$$R_{\min,k} = \{i | \nabla h_{t,k}(i) \nabla h_{t,k}(i+1) < 0 \wedge \nabla^2 h_{t,k}(i) > 0\} \tag{9}$$

where $R_{\min,k}$ is the set of minima for a scale k.

With regions spanned by minima and maxima of the target histogram available, we can now reshape each corresponding region of the source histogram independently. The bounds $[a, b]$ of a region j are given by $a = R_{\min,k}(j)$ and $b = R_{\min,k}(j+1)-1$. To reshape the source histogram H_s, we first compute the mean and standard deviations of each region:

$$\mu_{s,k}(j) = \sum_{i=a}^{b} \frac{h_{s,k}(i)}{b-a} \tag{10}$$

$$\sigma_{s,k}(j) = sqrt \sum_{i=a}^{b} \frac{(h_{s,k}(i) - \mu_{s,k}(j))^2}{b-a} \tag{11}$$

where $\mu_{s,k}(j)$ and $\sigma_{s,k}(j)$ are the mean and standard deviation of the jth region of $H_{s,k}$, respectively. The mean $\mu_{t,k}(j)$ and standard deviation $\sigma_{t,k}(j)$ of the target region are computed similarly.

The bins of corresponding regions are then reshaped by

$$h_{o,k}(i) = (h_{s,k}(i) - w_{s,k}\mu_{s,k}(j)) \frac{w_{t,k}\sigma_{t,k}(j)}{w_{s,k}\sigma_{s,k}(j)} + w_{t,k}\mu_{t,k}(j) \tag{12}$$

where $h_{o,k}$ is the set of output histogram bin counts for a given scale k and $w_{s,k}$ is a weight dependent on k, with $w_{t,k} = 1 - w_{s,k}$. The weight $w_{s,k}$ (and $w_{t,k}$) offers additional control over the amount of matching between the source and the target.

As one of the aims of our approach is to minimize the amount of user input required, we examined the interaction between different values for the weight parameter and the number of scales used in the matching. Fig. 5 shows the effect of different weights in conjunction with varying numbers of iterations for one example. We choose to set $w_{s,k} = k/S_{\max}$ as this consistently leads to a linear progression between the source and the target images. In Fig. 5 this is shown along the diagonal. In addition to allowing for a linear progression, setting the weight parameter in this manner simplifies the user's control.

Once all regions for a given scale k are matched, we compute the minima of the now updated $h_{o,k}$. We then apply an additional match of means and standard deviations, this time on the updated histogram $h_{o,k}$. The first transfer between histograms takes into account the features of the target, while the second transfer considers features of the source that have not yet been successfully reshaped by the first transfer. This ensures that even significantly different histograms will be reshaped and aligned correctly. The updated histogram is then used as the source for the next scale.

As we are reshaping regions based on the mean and standard deviation, we implicitly assume that these regions are Gaussian. In an ideal scenario, where the source and target images are very similar, features of the two histograms would naturally align and thus this Gaussianity assumption would hold for both source and target regions simultaneously. However, if minima-bounded regions in the source and target are not aligned, this assumption only holds for one of the histograms at a time. As such, by considering regions based on both the source and the target, we ensure that eventually corresponding features are created.

Fig. 4 shows resulting histograms of the matching process at each scale. The first scale matched is the coarsest (top left), where the shape of the target is simplified to a small number of peaks. Due to the small weight used for the first scales, the intermediate resulting histograms (shown in blue) remain close to the original. Further iterations do however gradually reshape the histogram closer to the target.

The only user parameter required by our algorithm is a percentage defining the number of scales that should be used.

The histogram reshaping is performed in coarser scales first and finer scales are only considered when the user requests a closer match. As a result, a lower percentage will only reshape the source histogram so that it matches persistent features of the target histogram. For instance, if the user requests a 20% match, then only a set of scales $k=\{1,\ldots,0.2S_{\max}\}$ is used.

After the source histogram is reshaped, the output image I_o is created through full histogram matching between the source image I_s and the reshaped histogram H_o. The cumulative histograms of the source and output C_s and C_o are used in this computation:

$$C_s(j)=\sum_{i=1}^{j} h_s(i), \quad j=1,\ldots,B \tag{13}$$

$$C_o(j)=\sum_{i=1}^{j} h_o(i), \quad j=1,\ldots,B \tag{14}$$

$$I_o(p)=v_o\left(C_o^{-1}\left(C_s\left(\frac{I(p)-\min(I)+1}{V}\right)\right)\right) \tag{15}$$

where a cumulative histogram C is defined as a function mapping a bin index to a cumulative count. The inverse function C^{-1} acts as a reverse lookup on the histogram, returning the bin index corresponding to a given count.

The nature of our algorithm allows it to naturally extend to pairs of high dynamic range images, or a combination of low and high dynamic range images. An example of partial matches between two HDR images is shown in Fig. 9, while Fig. 10 compares color transfers between pairs of HDR images for several different algorithms. As both the source and target images are HDR, the resulting image also has a high dynamic range. As such all images were tonemapped with the photographic operator [23] for visualization purposes, using the same parameters across all images. Fig. 9 demonstrates that a natural progression between pairs of HDR images can be achieved, despite differences in dynamic range. Further, Fig. 10 shows natural and adequate transfers may be difficult to achieve with other algorithms.

3.3. Detail control

Our approach transfers the color palette between images by manipulating their histograms. This affords high computational efficiency since it allows us to operate in a 1D space. At the same time, if the images chosen are of low quality or highly compressed, or if an area of the image with a smooth gradient changes to very different colors, artifacts can appear in the final result. These take the form of enhanced noise or compression artifacts. Previous solutions to this class of problems in the context of color transfer include the de-graining approach proposed by Pitié et al. [6] and the gradient-preservation step proposed by Xiao and Ma [4]. In both these approaches, an optimization step is used to modify the gradients of the resulting image so that they approach the gradients of the original source.

Both approaches achieve the desired result but the optimization step can significantly affect the time performance of the

Fig. 9. This is an example of partial matches between high dynamic range images. Only matches up to 50% are shown here as in this particular example, further scales did not produce significantly different results.

Fig. 10. Both source and target images have a high dynamic range. Our algorithm seamlessly handles pairs of arbitrary dynamic ranges. Existing algorithms however can lead to unexpected results as shown here.

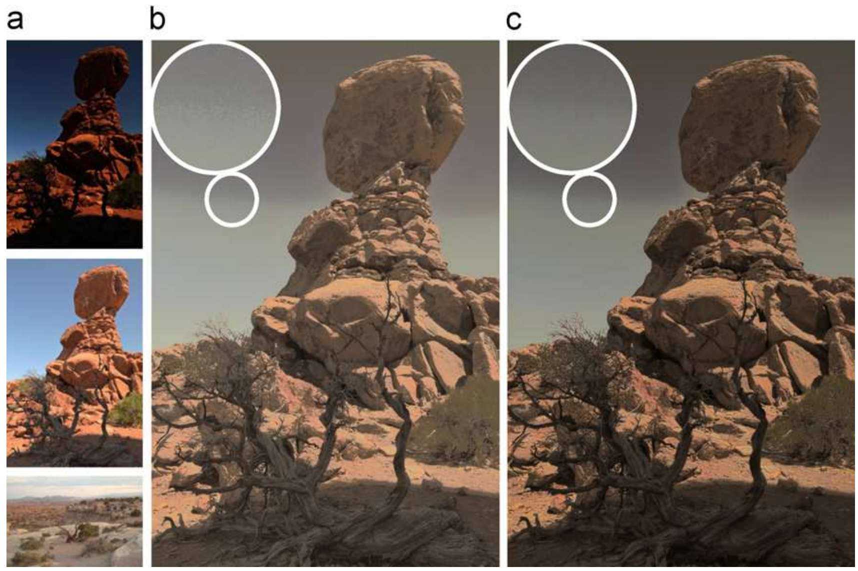

Fig. 11. The result (b) was produced by simply matching the source HDR image to the shown LDR target (both shown in (a)). In this case, no contrast enhancement or detail modification was applied to the image. As can be seen in the zoomed-in detail, noise in the sky region has been amplified and is now visible. By adding the difference in detail as shown in Eq. (17) this extraneous detail is removed, while missing detail in other regions of the image, such as the rocks, has been added. This is shown in (c).

algorithms. Moreover, halo artifacts can appear (an example is shown later in the paper in Fig. 20(h)). We propose a simpler and more efficient approach to counter such problems. Enhanced noise or compression artifacts can become more apparent in the transformed image because the contrast in these areas changes. As these are generally local changes, we have found that a sufficient solution is to manipulate local contrast in the resulting image such that it resembles the original local contrast.

The bilateral filter is a natural solution to this problem as it respects global edges but can filter over smaller local details. Thus, to modify local contrast in the output image, we manipulate the residual after subtracting the filtered image. As given in Eq. (6), I_{bilat} is the result of filtering an image I with the bilateral filter [31]. We define the residual as:

$$I_{\text{res}} = I - I_{\text{bilat}}. \tag{16}$$

To obtain the contrast-modified version I_o' of the output image, we consider both the detail of the original source $I_{\text{res},s}$ and the output $I_{\text{res},o}$:

$$I_o' = I_o + w_c(I_{\text{res},s} - I_{\text{res},o}). \tag{17}$$

The w_c parameter allows some control over the amount of correction applied to the detail layer of the image. In most LDR transfers we have experimented with this step was not necessary ($w_c = 0$) but in cases where the smooth areas of the input are particularly noisy, a value of 1 for w_c produced satisfying results. Fig. 11 demonstrates the effect of this step. This process is carried out for each channel separately.

4. Region selection

The aim of color transfer is to match the color palette of the target image. Occasionally however, additional control may be required to specify which parts of the source image should be recolored or which parts of the target should be considered. As discussed in Section 2, several different solutions have been proposed to create an influence map, using strokes, patches or swatches.

In this work, we use alpha mattes created using the Soft Scissors approach [33], although any technique or set of tools that can create an influence map could be substituted. A matte can be provided for the source image, the target or both. In the case of the source, the matte defines regions of the image that should not be changed and can be in the form of a binary mask, a soft selection or a fuzzy influence map. Similarly, the target mask defines which parts of the image should contribute to the color transfer. Fig. 6 shows an example where a matte was provided for both the source and the target.

Another case where more specific manual control may be required is when achromatic parts of the source image acquire a particular hue. This stems from the fact that decorrelated color spaces are decorrelated only for ensembles, while individual images may show significant correlations between the three channels. An example can be seen in Fig. 7(c) where the white tiger has acquired a distinctive red tint.

Although a matte could still be used in this case, to ensure that achromatic regions of the image look natural, we propose the following simple solution which anchors these regions of the histogram while allowing for the rest of the histogram to change. Note that achromatic colors are represented in CIELab with a value of 0 in both the a and b channels. A simple mask M is computed for each of the two chromatic channels for each pixel p to detect which pixels are achromatic in the input image before its colors are matched to the target:

$$M(p) = \begin{cases} 1 & |I(p)| > w_a(\max(I) - \min(I)) \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where w_a is a constant defining the percentage of the range of each channel that should be considered. We empirically chose a value of 0.08 for w_a throughout our experiments as it captured the achromatic region of the histogram without including chromatic information. Example masks for different values of w_a are shown in Fig. 7.

After the histogram reshaping, the pixel values of regions in the image that were previously achromatic will have shifted by some amount. Using the mask computed in Eq. (18) the displacement for each previously achromatic pixel can be determined. The mask is then convolved with a Gaussian filter kernel to ensure a smoother result near the edges of the masked regions and is scaled by the displacement of the achromatic values after the histogram reshaping. Finally, the mask is applied to the image, correcting for unwanted color shifts. Fig. 7(d) demonstrates the effectiveness of histogram anchoring. Note that now the tiger has remained white.

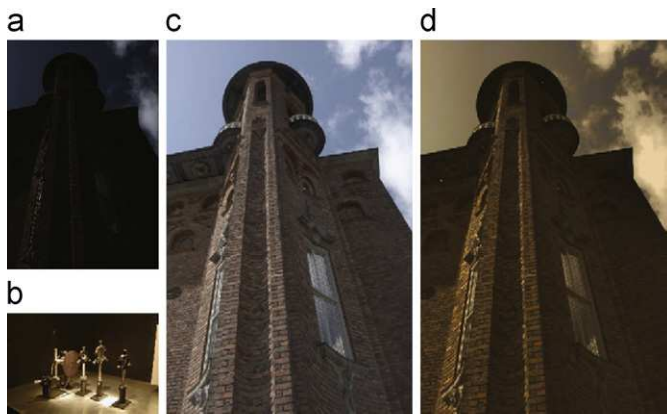

Fig. 12. Another example of and HDR to LDR match. Here both the dynamic range and the colors of the target are very different to the source. The result has successfully matched both. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) (a) Source, (b) target, (c) Reinhard et al. [1], (d) our result.

5. Creative tone reproduction

Using our technique it is possible not only to transfer the color palette between two images but also to reduce the dynamic range of the source image to match that of the target. Fig. 14 demonstrates the tonemapping capabilities of our algorithm for a selection of images. As can be seen, our histogram reshaping technique successfully compress the histogram of the high dynamic range (HDR) source image. This is achieved as follows.

The source and target images are first converted to the CIELab color space and transformed according to the steps presented in Section 3. This color space, in addition to decorrelating the data in the image, also involves a cube root compression step [34]. This by itself is not an effective tonemapping operator but it helps condition the HDR histogram so that more useful information can be captured with fewer bins. The reshaping of the histograms, in addition to matching the colors between the two images also further compresses contrasts to match those of the LDR target.

Due to the compression of the matching step, local contrast in the image can be reduced. To this end, we modify the contrast adjustment step presented in Section 3.3 to ensure that useful local detail is preserved while artifacts are not enhanced. We first compute the detail layer of the HDR source using Eq. (16). This is then modified by the detail of the output image and the result is added to the output to enhance the local detail (Eq. (17)). Since the

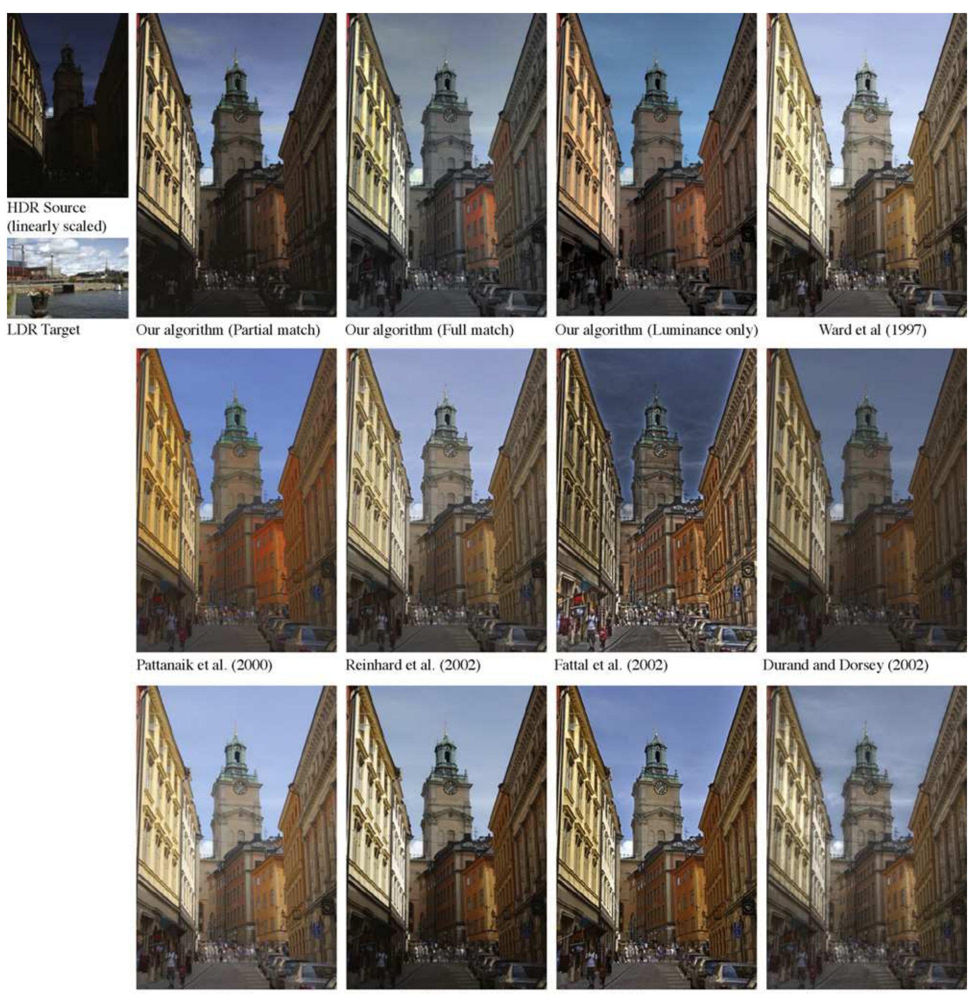

Fig. 13. Here, a partial and a full match are shown for an HDR source and an LDR target. The partial match is much closer to the source both in colors and contrasts. The full match on the other hand shows much more detail. The result of a luminance only transfer is also shown and compared with corresponding results using several tonemapping algorithms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

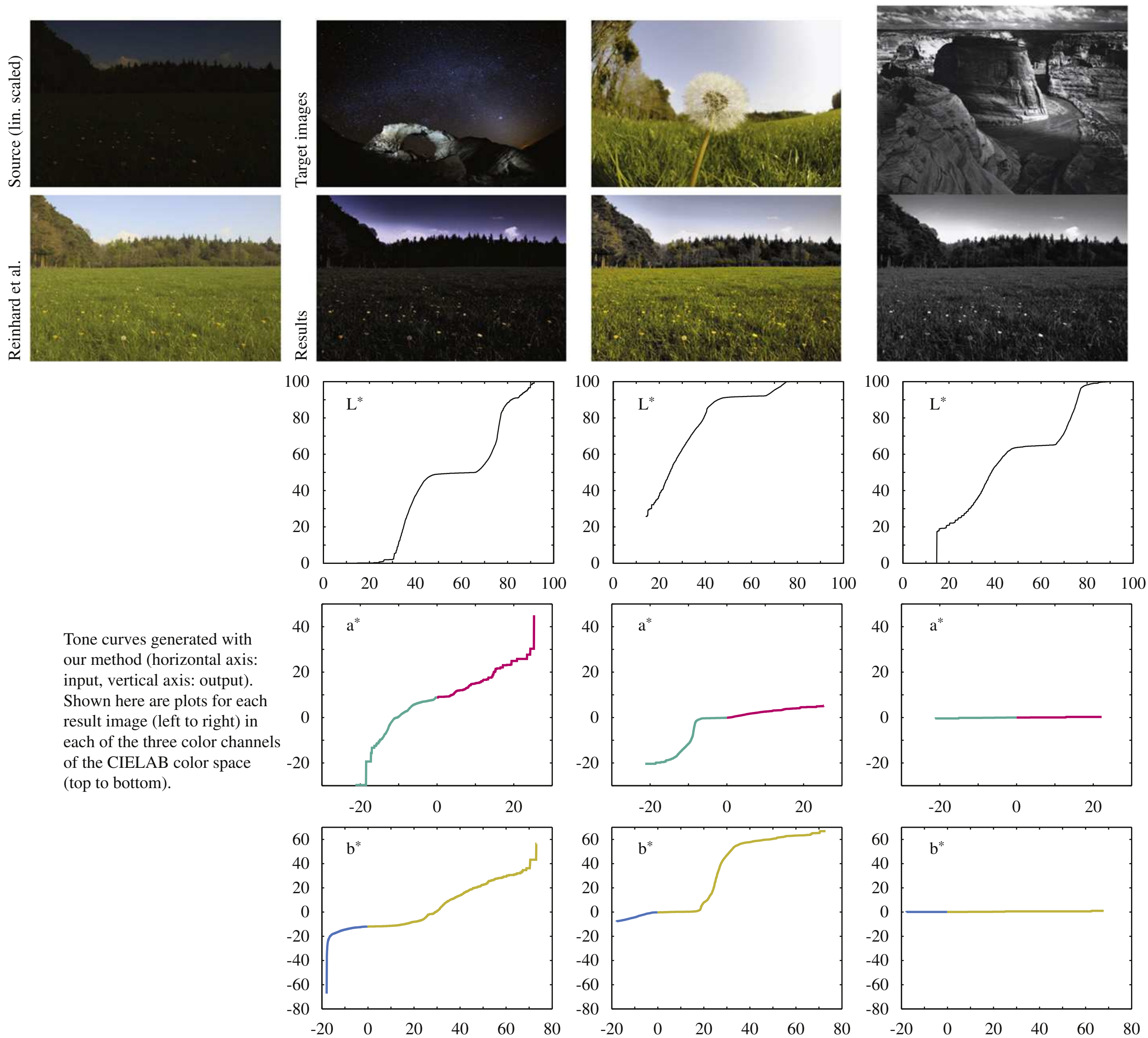

Fig. 14. Tone reproduction examples created using our technique. Here the same source image was mapped to various different targets to get different appearances. The resulting tone curves are shown for each channel. The night image used as a target is courtesy of Dan Ransom (http://www.danransom.com).

contrast in the original source HDR image is much higher than the local contrast in the compressed output, this step is able to both enhance the contrast of desirable features while suppress details that were made visible due to the color transfer.

To ensure that partial matching is still possible when matching an HDR and an LDR image, the contrast can be weighted by the desired percentage c of matching that is used for the main part of our algorithm, so the contrast adjustment parameter is set to $w_c = cS_{\max}$. Fig. 16 demonstrates the results of partial matching for a range of levels between the HDR source and LDR target of Fig. 15.

6. Results

Our approach transfers the color palette between images of arbitrary dynamic ranges, combining the areas of color transfer and tone reproduction. For that purpose, we evaluate our technique against representative algorithms from both these areas.

6.1. Color transfer

To evaluate the color transfer capabilities of our technique, we compare our results with three representative color transfer algorithms. We have chosen the color transfer technique by Reinhard et al. [1], the ND-pdf transfer by Pitié et al. [5] and finally, the more recent method by Xiao and Ma [4]. Fig. 21 shows the results of the different methods for a selection of images. For each pair of images, a 50% and a 100% match produced by our algorithm is shown.

Our results show a clear progression between the appearance of the source and that of the target image. In the first example, a 50% match turns most of the tree leaves yellow while a 100% match successfully transfers the color palette of the target image fully.

Fig. 15. The source and target images used for the results in Fig. 16.

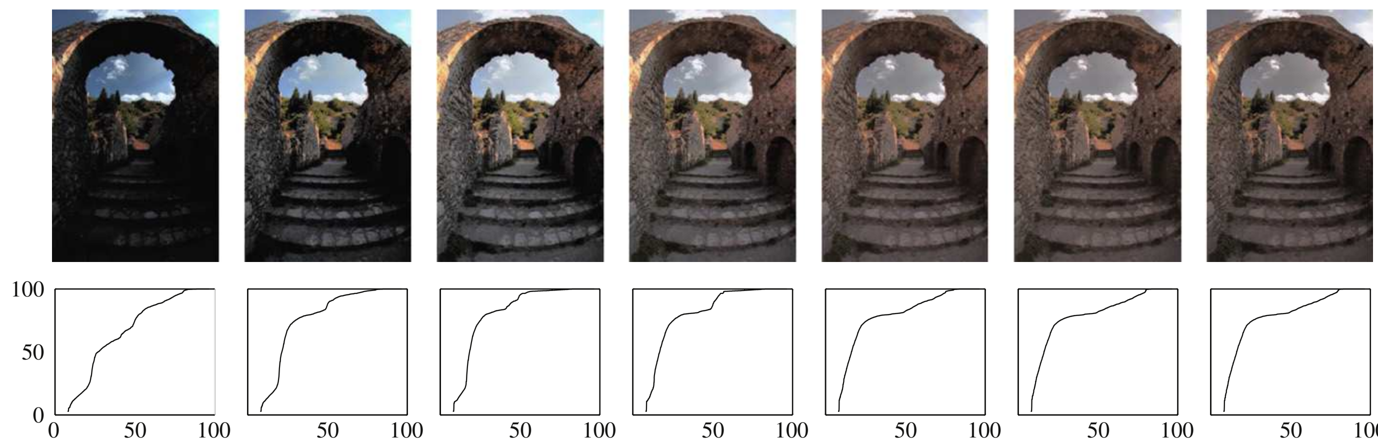

Fig. 16. A series of partial results was created using the images shown in Fig. 15. The following values were used for the matching parameter (going from left to right): 15%, 25%, 35%. 50%, 65%, 80% and 100%. For each of these results, the corresponding tone curve mapping input to output pixel values was computed. A clear progression can be seen, leading to the final tone curve. Matches closer to the source result in a curve close to a 45° line.

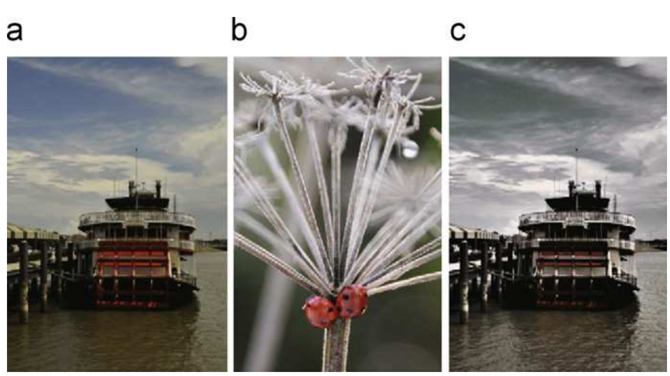

Fig. 17. Here, the effect achieved on the resulting image approaches selective desaturation as the target is largely achromatic.

Note also that the corresponding results for all other methods shown in this particular case have assigned yellow or red tints to the sky region, giving the resulting images an unnatural look. The anchoring step described in Section 4 was used in this case to counter this effect.

Similar observations can be made for the other examples shown. In the fourth example of Fig. 21 in particular (Utah—Night), the progression from 50% to the 100% match has successfully increased the contrast and reduced the saturation of the image approaching the appearance of the given target. Although the color rendition produced by Pitié et al. [5] correctly matches the color palette of the target, noticeable spatial artifacts are present in the sky. On the other hand, the result by Xiao and Ma [4] creates halo artifacts near extreme edges.

Our algorithm is particularly suitable for partial or complete desaturation. As shown in Figs. 5 and 18, the result has successfully matched the target Ansel Adams photograph in terms of both color palette and contrasts. Fig. 19 shows the histograms of the red-green channel for the source, target and output images of Fig. 5a, b and f. Our algorithm has successfully suppressed the chromatic channels. Although the result from Reinhard et al. [1] has also rendered the image grayscale, it is clear that the overall appearance of the target is not matched. A less extreme scenario is demonstrated in Fig. 17.

6.2. Tone reproduction

To demonstrate the behavior of our algorithm in the tone reproduction cases, we computed the tone curves for each of the transfers shown in Fig. 14. The three curves for each of the images show how the values for each channel are mapped from the source to the resulting image. As can be seen, our algorithm allows for a non-linear mapping of luminance values, dependent on the target image, which is necessary for effectively reducing the dynamic range while still preserving details in the image. Moreover, when a grayscale target is used, the tone curve of the chromatic channels becomes flat, as would be expected since all values are mapped to zero.

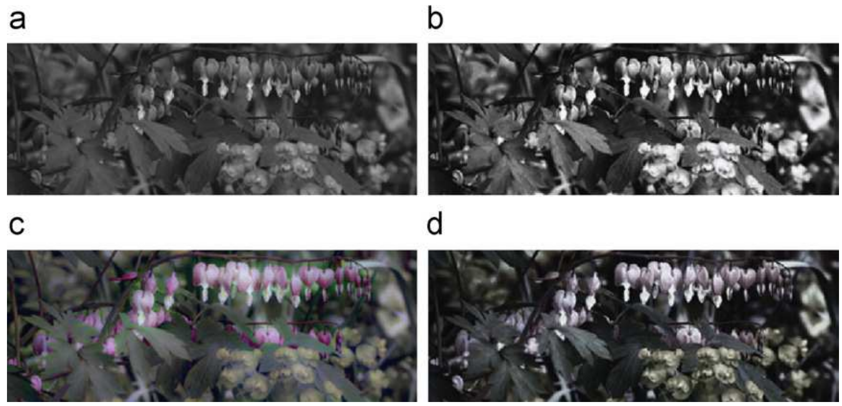

Fig. 18. Results of the same source and target images as Fig. 5 using other color transfer techniques.

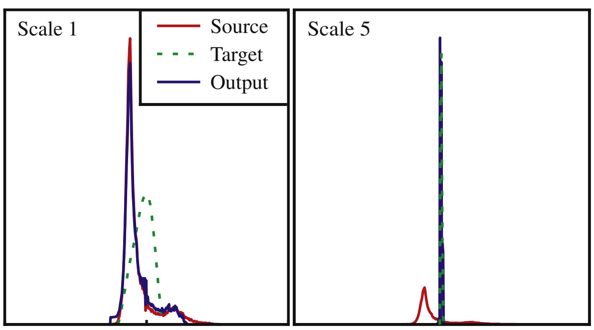

Fig. 19. The coarsest and finest scale histograms for the source, target and fully matched result from Fig. 5(a), (b) and (f). The source histogram is successfully reshaped to match the target for both of the chromatic channels, rendering the image to grayscale.

To further evaluate the progressive tone reproduction results, the tone curve mapping source to output intensities was computed for each of the partial matches (also shown in Fig. 16). As can be seen, matches closer to the source result in a tone curve close to a 45° line: pixel values in the image are almost unchanged. However, using further scales leads to a more compressive tonemapping curve and a match more closely resembling the target in both color content and intensities.

Additional results of the tone reproduction capabilities of our algorithm can be seen in Figs. 12, 14 and 20. Further, Fig. 13 shows a partial and a full match as well as a luminance-only match. The results for several tonemapping operators are also shown, using the source image as the input. Our algorithm successfully compresses the dynamic range of the source image while preserving local contrast, producing results comparable to the state of the art. Dynamic range differences between the source and target images are successfully handled, producing results comparable with state-of-the-art tonemapping techniques. The color transfer methods compared in Fig. 20(e)–(h) are designed for palette transfers between images of similar dynamic ranges and consequently produce unexpected results in this case.

All the results for this section were produced on an Apple MacBook Pro with a 2.4 GHz Intel Core 2 Duo CPU and 2 GB of RAM running Snow Leopard and Matlab v7.6.0. A Matlab version of all the techniques tested was acquired for comparison and the relative timings between the different algorithms tested can be found in Table 1. Note that the time performance of our technique depends on the number of bins selected. For all the examples shown, 400 bins were used unless otherwise mentioned. As can be seen, the time performance of our algorithm is on a par with most other techniques.

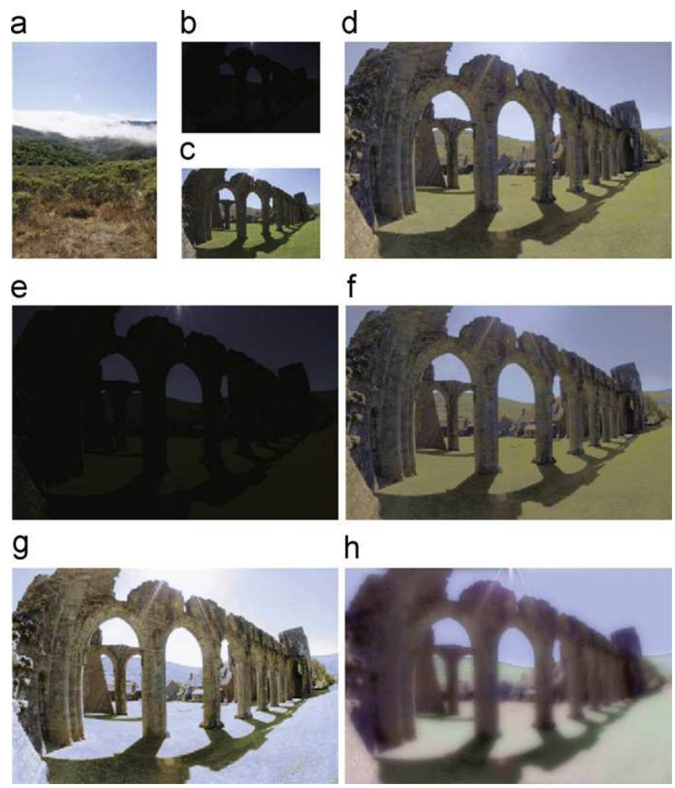

Fig. 20. The target chosen here has a similar color palette to the source image but much lower dynamic range. The result has matched the dynamic range of the target, producing comparable results to existing tonemapping techniques. For display purposes, we show the linearly scaled source (b) and a tonemapped version (c) using [23]. Images (e)–(h) were created using other color transfer techniques. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) (a) Source, (b) target, (c) tonemapped, (d) our method (100% match), (e) Reinhard et al. [1], (f) histogram matching, (g) Pitió et al. [5], (h) Xiao and Ma [4].

6.3. Limitations

We have found the method to be relatively robust. However, it does make some underlying assumptions with associated

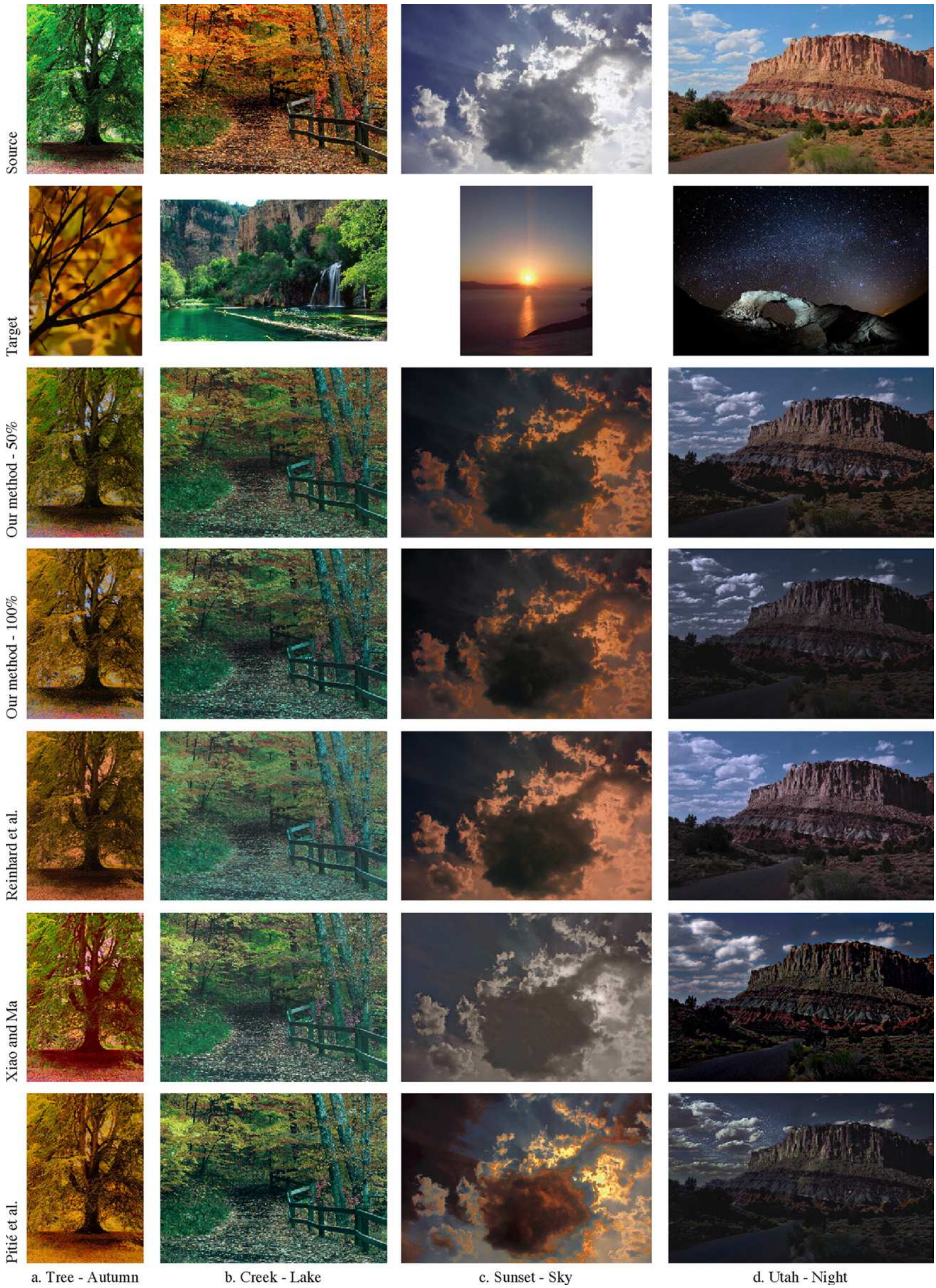

Fig. 21. Comparisons with existing methods. Lake image: ''Hanging Lake'' by Brent Reed, brent@reedservices.com, adopted with permission of the artist. Creek image: ''Mc. Cormic Creek State Park, Indiana'' by Mike Briner, mbphoto@sprynet.com, www.mikebrinerphoto.com, adopted with permission of the artist.

Table 1
Relative timings between the tested algorithms, computed over the image pairs shown in Fig. 21. The performance of our algorithm for a full match over 400 bins has been taken as the baseline for these comparisons. Lower numbers indicate better performance and vice versa.

Method	Relative performance
Ours (100%—400 bins)	1.000
Ours (50%—400 bins)	1.084
Ours (100%—255 bins)	0.687
Reinhard et al.	0.026
Pitié et al.	1.626
Xiao and Ma	2.538

limitations. In particular, we assume that the three color channels are decorrelated for the given inputs. Although the CIELab color space is on average decorrelated, this is not necessarily the case for individual images. For many images this is in practice not a problem. For other specific cases, as shown earlier in Fig. 7, the result can be corrected by our histogram anchoring technique. However, there do remain some cases whereby channel correlations prevent a sensible result, as with other techniques that assume channel independence.

A possible solution to this problem would be to compute a color space using the approach proposed by Ruderman et al. [2], which would maximally decorrelate the given images. A number of existing color transfer techniques have experimented with this approach [7–10]. Nonetheless, we have found that, for the purposes of our algorithm, CIELab sufficiently decorrelates for a wide range of images, as demonstrated throughout the paper.

7. Summary

We propose a novel color transfer method that offers the ability to progressively match the color palette of a given target image. We achieve this using a histogram reshaping method that is capable of matching the features of the target histogram to a selected level of accuracy to yield a variety of creative effects. The progressive nature of the algorithm allows partial transfers, inducing more subtle effects than would be possible with classical color transfer algorithms and histogram matching techniques. We also find that the method is comparatively robust and computationally efficient, which can be attributed to the fact that most operations are performed on the histograms of pairs of images. The improved robustness allows a wider range of target images to be used, and thereby increases the artistic freedom it affords.

Our method uniquely allows artistic tone reproduction by matching a high dynamic range image against an appropriately chosen low dynamic range reference image. Combined with the use of straightforward parameters, in particular a slider that specifies the percentage of transfer, this broadens the appeal of the technique to digital artists and photographers.

Acknowledgements

We would like to thank Douglas Cunningham for all the useful suggestions and discussions.

References

[1] Reinhard E, Ashikhmin M, Gooch B, Shirley P. Color transfer between images. IEEE Computer Graphics and Applications 2001;21:34–41.
[2] Ruderman D, Cronin T, Chiao C. Statistics of cone responses to natural images: implications for visual coding. Journal of the Optical Society of America A 1998;15(8):2036–45.
[3] Neumann L, Neumann A. Color style transfer techniques using hue, lightness and saturation histogram matching. In: Computational Aesthetics in Graphics Visualization and Imaging 2005:111–22.
[4] Xiao X, Ma L. Gradient-preserving color transfer. Computer Graphics Forum 2009;28:1879–86.
[5] Pitié F, Kokaram A, Dahyot R. N-dimensional probability density function transfer and its application to colour transfer. In: ICCV '05: Proceedings of the 2005 IEEE international conference on computer vision, vol. 2. Washington, DC, USA: IEEE Computer Society; 2005. p. 1434–9.
[6] Pitié F, Kokaram A, Dahyot R. Automated colour grading using colour distribution transfer. Computer Vision and Image Understanding 2007;107(2):1434–9.
[7] Abadpour A, Kasaei S. A fast and efficient fuzzy color transfer method. In: Proceedings of the 4th IEEE international symposium on signal processing and information technology. Washington, DC, USA: IEEE Computer Society; 2004. p. 491–4.
[8] Abadpour A, Kasaei S. An efficient PCA-based color transfer method. Journal of Visual Communication and Image Representation 2007;18:15–34.
[9] Grundland M, Dodgson N. The decolorize algorithm for contrast enhancing, color to grayscale conversion. Technical Report UCAM-CL-TR-649, University of Cambridge; 2005.
[10] Xiao X, Ma L. Color transfer in correlated color space. In: VRCIA '06: Proceedings of the 2006 ACM international conference on virtual reality continuum and its applications. New York, NY, USA: ACM; 2006. p. 305–9.
[11] Chang Y, Uchikawa K, Saito S. Example-based color stylization based on categorical perception. In: Proceedings of the 1st symposium on applied perception in graphics and visualization APGV 2004. New York, NY, USA: ACM; 2004. p. 91–8.
[12] Welsh T, Ashikhmin M, Mueller K. Transferring color to greyscale images. In: Proceedings of the 29th annual conference on computer graphics and interactive techniques, vol. 21. New York, NY, USA: ACM; 2002. p. 277–80.
[13] Ji Y, Liu H, Wang X, Tang Y. Color transfer to greyscale images using texture spectrum. In: Proceedings of 2004 international conference on machine learning and cybernetics; 2004. p. 4057–61.
[14] Yan W, Kankanhalli M. Colorizing infrared home videos. In: Proceedings of the 2003 international conference on multimedia and expo, vol. 1. Washington, DC, USA: IEEE Computer Society; 2003. p. 97–100.
[15] Levin A, Lischinski D, Weiss Y. Colorization using optimization. ACM Transactions on Graphics 2004;24(3):689–94.
[16] Wen CL, Hsieh CH, Chen BY, Ming O. Example-based multiple local color transfer by strokes. Computer Graphics Forum 2008;27(7):1762–5.
[17] Maslennikova A, Vezhnevets V. Interactive local color transfer between images. In: GraphiCon; 2007.
[18] An X, Pellacini F. Computer Graphics Forum 2010;29:263–71.
[19] Senanayake C, Alexander D. Colour transfer by feature based histogram registration. In: British machine vision conference. British Machine Vision Association; 2007.
[20] Reinhard E, Ward G, Pattanaik S, Debevec P. High dynamic range imaging: acquisition, display and image-based lighting. San Francisco: Morgan Kaufmann Publishers; 2005.
[21] Drago F, Myszkowski K, Annen T, Chiba N. Adaptive logarithmic mapping for displaying high contrast scenes. Computer Graphics Forum 2003;22(3):419–26.
[22] Ward G, Rushmeier H, Piatko C. A visibility matching tone reproduction operator for high dynamic range scenes. IEEE Transactions on Visualization and Computer Graphics 1997;3(4):291–306.
[23] Reinhard E, Stark M, Shirley P, Ferwerda J. Photographic tone reproduction for digital images. ACM Transactions on Graphics 2002;21(3):267–76.
[24] Durand F, Dorsey J. Fast bilateral filtering for the display of high-dynamic-range images. ACM Transactions on Graphics 2002;21(3):257–66.
[25] Fattal R, Lischinski D, Werman M. Gradient domain high dynamic range compression. ACM Transactions on Graphics 2002;21(3):249–56.
[26] Li Y, Sharan L, Adelson E. Compressing and companding high dynamic range images with subband architectures. ACM Transactions on Graphics 2005;24(3):836–44.
[27] Bae S, Paris S, Durand F. Two-scale tone management for photographic look. ACM Transactions on Graphics 2006;25(3):637–45.
[28] Farbman Z, Fattal R, Lischinski D, Szeliski R. Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Transactions on Graphics 2008;27(3):67(1)–10).
[29] Lischinski D, Farbman Z, Uyttendaele M. Interactive local adjustment of tonal values. ACM Transactions on Graphics 2006;25(3):646–53.
[30] Hertzmann A, Jacobs C, Oliver N, Curless B, Salesin D. Image analogies. In: SIGGRAPH '01: Proceedings of the 28th annual conference on computer graphics and interactive techniques. New York, NY, USA: ACM; 2001. p. 327–40.
[31] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: ICCV '98: Proceedings of the 1998 IEEE international conference on computer vision. Washington, DC, USA: IEEE Computer Society; 1998. p. 839–46.
[32] Chen J, Paris S, Durand F. Real-time edge-aware image processing with the bilateral grid. ACM Transactions on Graphics 2007;26(3):103–1–9.
[33] Wang J, Agrawala M, Cohen MF. Soft scissors: an interactive tool for realtime high quality matting. ACM Transactions on Graphics 2007;26(3). 9(1)–(6).
[34] Reinhard E, Khan EA, Akyüz AO, Johnson GM. Color imaging: fundamentals and applications. Wellesley: AK Peters; 2008.