

Correction of Clipped Pixels in Color Images

Di Xu, *Student Member, IEEE*, Colin Doutre, *Student Member, IEEE*, and Panos Nasiopoulos, *Member, IEEE*

Abstract—Conventional images store a very limited dynamic range of brightness. The true luma in the bright area of such images is often lost due to clipping. When clipping changes the R, G, B color ratios of a pixel, color distortion also occurs. In this paper, we propose an algorithm to enhance both the luma and chroma of the clipped pixels. Our method is based on the strong chroma spatial correlation between clipped pixels and their surrounding unclipped area. After identifying the clipped areas in the image, we partition the clipped areas into regions with similar chroma, and estimate the chroma of each clipped region based on the chroma of its surrounding unclipped region. We correct the clipped R, G, or B color channels based on the estimated chroma and the unclipped color channel(s) of the current pixel. The last step involves smoothing of the boundaries between regions of different clipping scenarios. Both objective and subjective experimental results show that our algorithm is very effective in restoring the color of clipped pixels.

Index Terms—Clipping, desaturation, color restoration, high dynamic range (HDR), inverse tone mapping.



1 INTRODUCTION

RECENTLY, high dynamic range (HDR) displays have gained significant interest in industry. It is known that the dynamic range of conventional displays is very limited compared to the range of intensities perceivable by the human visual system. Conventional displays usually have a dynamic range of about two orders of magnitude, whereas the human visual system has an overall dynamic range of nearly 10 orders of magnitude, and can simultaneously perceive intensities over a range of about five orders of magnitude [1], [2]. This has motivated the development of HDR image capturing and display technology.

One important HDR imaging problem is that, given the HDR content, how to properly display them on a conventional low dynamic range (LDR) media. Tone mapping addresses this problem by reducing strong contrast from the HDR scene values to the LDR displayable range while preserving the image details and color appearance that are important to appreciate the original scene content. In recent years, various tone mapping operators [3], [4], [5], [6], [7], [8], [9], [10] have been developed in order to show the HDR contents on a conventional LDR display device or print.

Recently developed HDR displays [11] have greatly extended the limited dynamic range of conventional cathode ray tube (CRT), liquid crystal display (LCD), and projector-based displays. Akyüz et al. [12] show that HDR displays produce pictures with more appealing subjective quality than conventional LDR displays. In order to effectively display legacy LDR images and videos on HDR displays, inverse tone mapping schemes have been developed to extend the dynamic range of LDR images and videos. Legacy

images and videos store only a small dynamic range of information due to the limitations of the capturing and display devices. The very bright or dark parts of a scene are clipped to the upper or lower displayable limits, and as a result, information is lost. Special attention has been paid to the restoration of the clipped pixels. Over the last few years, several methods [13], [14], [15], [16], [17] have been developed to enhance the luma of the clipped pixels so that the enhanced clipped regions have higher dynamic range and look more realistic on HDR displays.

Meylan et al. [14] apply a simple piecewise linear tone scale function, composed of two slopes, one applied to the diffuse areas and one applied to the specular reflected areas, in order to particularly enhance the specular highlights. In [15], the Median Cut algorithm and Gaussian filter are applied to estimate the light source density map. Then a luma expansion is applied to enhance the clipped regions. In [16], a smooth brightness enhancement function is obtained by blurring a binary mask with a large kernel of approximately Gaussian shape. A semiautomatic classifier was developed in [17] to classify the clipped regions as lights, reflections, or diffuse surfaces. Each class of objects is enhanced with respect to its relative brightness. All of the above schemes enhance only the luma, while the chroma enhancement is not considered.

For a clipped pixel, often not all three red (R), green (G), and blue (B) channels are clipped, nor does the same amount of clipping occur in each channel. If clipping changes the R , G , B color ratios of a pixel, then the result is color distortion. In fact, color distortion occurs very often. Although people are accustomed to the clipping effect in highlights, where the distorted color is desaturated and close to white, the distorted colors near the midtone produce a very noticeable and disturbing effect. Fig. 1 gives an example of color distortion caused by the clipping effect. Fig. 1a is a correct exposure image, and Fig. 1c is the corresponding overexposed image, where the yellow pixels reveal obvious color distortions due to the clipping. We enhance the luma of the clipped image using the best possible values (i.e., the ground truth luma of the correct

• The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada. E-mail: {dixu, colind, panos}@ece.ubc.ca.

Manuscript received 27 May 2009; revised 29 Sept. 2009; accepted 27 Oct. 2009; published online 8 Apr. 2010.

Recommended for acceptance by G. Drettakis.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number TVCG-2009-05-0098. Digital Object Identifier no. 10.1109/TVCG.2010.63.

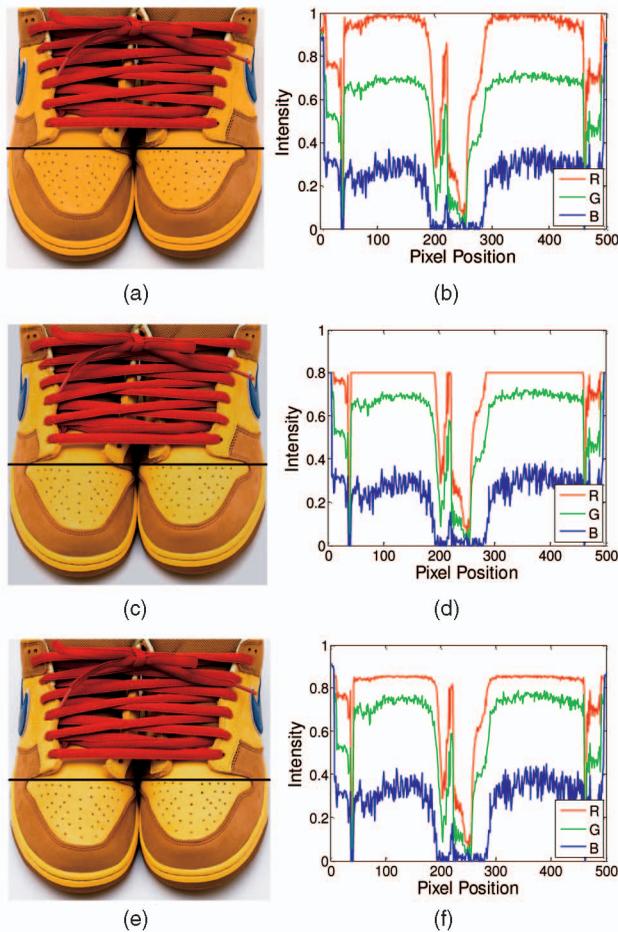


Fig. 1. Example of color distortion due to the clipping. (a) A correct exposure image, (c) the corresponding overexposed image, and (e) the enhanced image with corrected luma for clipped pixels. (b), (d), and (f) The plots of the RGB intensities versus the pixel index along the highlighted horizontal lines of images (a), (c), and (e), respectively.

exposure image), while keeping the chroma distortion unchanged. The luma-enhanced image is shown in Fig. 1e. Figs. 1b, 1d, and 1f depict the plots of RGB intensities versus the pixel position along the same horizontal line shown in Figs. 1a, 1c, and 1e, respectively. In Figs. 1c and 1d, we observe that the red channel is saturated for most pixels on the shoe (nonshoelace part); and for this reason, this part appears yellow in Fig. 1c rather than being orange as in the correct exposure picture (in Fig. 1a). As shown in Fig. 1e, enhancing only the luma does not solve the color distortion problem. In fact, the color distortion in this image appears at least as disturbing as in the clipped image (Fig. 1c). Ideally, instead of changing all the color channels as it is done by luma enhancement, we want to enhance only the clipped color channel(s). Fig. 1f shows that enhancing luma makes the unclipped channels (i.e., G and B) less accurate. Therefore, we need an algorithm that corrects color and at the same time enhances luma of a clipped image.

A few methods were developed to enhance the color saturation and fill in clipped regions. One category of the enhancement methods is to remove the specularity from the highlighted area and reconstruct the highlighted object assuming only diffuse reflection exists [18], [19], [20]. These methods are particularly useful as a preprocessing before

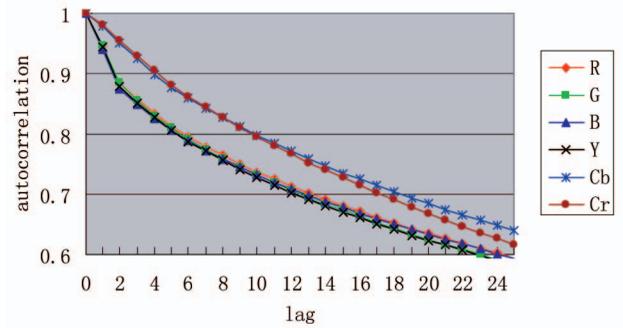


Fig. 2. Normalized autocorrelation of R , G , B , Y , Cb , and Cr signals (average over 24 true-color Kodak images).

tasks such as stereo matching, segmentation, and object recognition. The highlight-removed images, however, do not reflect the real scene under the original lighting. Hence, these approaches cannot restore the lost information due to clipping. Wang et al. [21] proposed an effective HDR image hallucination approach for adding HDR details to the overexposed and underexposed regions of an LDR image. The method assumes that high-quality patches exist in the image with similar textures as the regions that are over- or underexposed. It corrects both the luma and the chroma, while it fills in detailed textures for the overexposed and underexposed regions. This approach, however, is semiautomatic and it needs the user's input to identify textures that can be applied to fill in the underexposed or overexposed areas. This manual intervention is often undesirable.

An automatic method proposed by Zhang and Brainard [22] uses a statistical approach to fix saturation for overexposed pixels in color images. This work exploits the correlation between the responses of RGB color channels at each pixel, and estimates the clipped color channel(s) using the Bayesian algorithm based on the corresponding unsaturated color channel(s) and the prior distribution parameters. The method is proved to be effective for enhancing clipped pixels for images with a small saturated area; however, it is not as effective for images with large saturated areas.

In this paper, we propose an effective clipped-pixel-enhancing algorithm, which automatically restores both the luma and chroma of the clipped pixels. We exploit the strong correlation in chroma between saturated pixels and their surrounding unsaturated pixels. Experimental results show that our algorithm outperforms the Bayesian algorithm [22] in both objective and subjective quality evaluations.

The rest of the paper is structured as follows: Section 2 describes our proposed method. The experimental results are presented in Section 3. In Section 4, we conclude the paper and point out the potential applications of our method.

2 OUR PROPOSED METHOD

In this paper, we aim at restoring the lost information in overexposed color images based on the strong spatial correlation in the chroma channels. The YCbCr color space is designed so that the chroma channels will be smooth in local regions for most images. It has been shown that utilizing the smoothness property of chroma [23], [24], [25] is better than assuming luma is smooth [16], [21]. Fig. 2 shows the normalized autocorrelation of R , G , B , Y , Cb , and

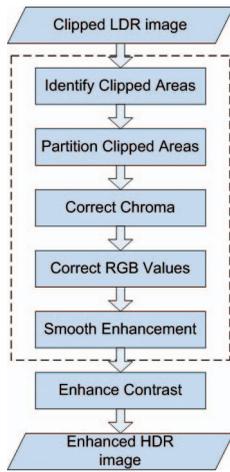


Fig. 3. Flowchart of our proposed method.

C_r at lags of 0-25 pixels. Each point in the figure is an average value over the 24 true-color Kodak images [26]. From the graph, we can see that there is stronger autocorrelation for the C_b and C_r channels than the R , G , B , and Y channels. Exploiting the strong spatial correlations in the C_b and C_r channels has more potential than exploiting the correlations in the R , G , B , or Y channels. For this reason, in our approach, we apply a chroma interpolation for the clipped pixels rather than directly correcting the R , G , and B signals.

Our proposed method can be broken down into several steps, which are shown in the flowchart in Fig. 3. First, we identify the clipped areas. Then, we partition each clipped area into smaller regions according to the chroma. We correct the chroma for each region and then correct the corresponding RGB values for all clipped pixels. Afterward, we apply a smoothing process to the corrected RGB values. The last step involves “Enhancing the Contrast” and can be performed by using any existing inverse tone mapping process [15], [16], [27], which is not the focus of our work. A detailed description of these steps is given in the following sections.

2.1 Identify Clipped Areas

Before doing any enhancement, we first need to identify the clipped areas. One way of doing this is to simply select pixels from all three color channels that have the maximum value (e.g., 255 for 8-bit per channel images). Fig. 4a shows a clipped image with a rectangular region of interest. Fig. 4b shows the clipped area (within the region of interest) identified with a simple threshold (the maximum value, e.g., 255 for 8-bit per channel images). The white pixels represent the clipped area. As it can be seen, this simple approach often generates very small isolated clipped areas and large clipped areas with small holes. The effect is due to image noise. The captured pixel values are determined not only by the light from the scene, but also by the camera response, sensor noise, and color filter array interpolation. The in-camera processing adds noise to a pixel value, and consequently, a clipped pixel may have a value slightly lower than the maximum value. For this reason, we first apply a bilateral filter [28], [29] to remove the noise. Then, a threshold τ is applied to each color channel of a pixel to

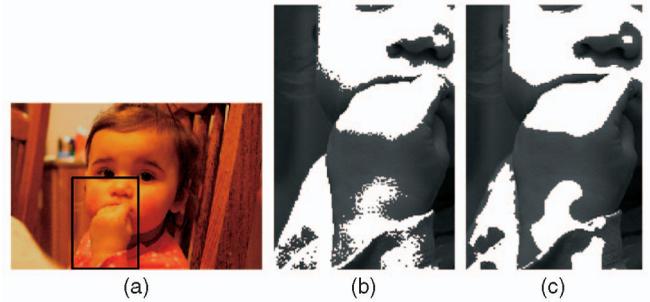


Fig. 4. Example of clipped areas. (a) Clipped image with a rectangular region of interest, (b) clipped areas in the region of interest identified with a simple threshold (R , G , or $B = 255$), and (c) clipped areas in the region of interest identified with our proposed method.

identify clipped pixels and channels. We experimentally selected τ to be 252.5 for 8-bit per channel images herein. Fig. 4c shows the clipped area in the region of interest identified by our method. We observe that the clipped area in Fig. 4c is quite clean and more appropriate for subsequent color correction compared to that in Fig. 4b. Note that the bilateral filter is used only for identifying the clipped areas. The original unfiltered image is used in all of the following steps to avoid losing detail from the image.

2.2 Partition Clipped Areas

The purpose of partitioning the clipped areas is to group the clipped pixels into regions with similar chroma before correcting the color for each region. We first partition the clipped areas into spatially disconnected regions, which probably belong to different surfaces and have different chroma. Each region may still contain multiple clipped objects that have different colors. We segment each region according to its chroma. To eliminate the illumination differences on the same color surface, we consider only chroma quantities, i.e., C_b and C_r , in the segmentation. For simplicity, in order to segment the region, we choose to use either the C_b or C_r , whichever has a larger variance within the considered region.

The pixels with heavier saturation (i.e., 2 or 3-channel saturated pixels) are considered as one subregion, where the color is potentially heavily distorted. The 1-channel saturated pixels are further segmented using a histogram-based multithreshold algorithm presented in [31]. This segmentation algorithm often results in a few large and many small subregions. Finally, we merge these small regions or the regions without a valid surrounding region (note: surrounding region will be explained shortly in Section 2.3) with their neighboring clipped regions. If more than one neighboring clipped region exists, the current region is merged with the neighboring region that is the closest to the current region in chroma.

An example of clipped area partition is given in Fig. 5, where connected clipped areas (i.e., white pixels in Fig. 5a) are partitioned into smaller regions (shown and numbered in Fig. 5b). Each region has similar chroma. This partitioning is essential for the subsequent color correction steps.



Fig. 5. Example of clipped area partition. (a) The clipped areas before partition (white pixels) and (b) the clipped areas partitioned into regions with similar chroma.

2.3 Correct Chroma

As explained before, the R , G , and B values are strongly affected by the illumination of the area. There are much stronger spatial variations in R , G , and B values than in chroma. Although the RGB and chroma in clipped areas can both be estimated using an interpolation method given their neighboring unclipped areas, a smooth signal, like the chroma, may be more accurately estimated, since there is less spatial variation associated with such a signal. For this reason, we chose to estimate the chroma values in a clipped region by smoothly interpolating the chroma of neighboring unclipped pixels. Once the chroma values are estimated, then we use them to calculate the corrected R , G , or B values in the clipped regions.

In order to correct a clipped region, we first attempt to find an unclipped region with similar chroma next to it. We select neighboring pixels with similar color to the clipped region as seed points. This is done by first choosing the unclipped or already corrected neighboring pixels with gradients of both chroma channels less than a threshold (experimentally, we determined 2.5 works well). Then, starting from each seed point, we apply a region growing algorithm shown in [31] to both C_b and C_r , and the intersection of the two obtained regions is a surrounding region with similar chroma to the clipped region. Since there may be small chroma variations within each clipped region, we take the union of all surrounding areas obtained from different seed points as the surrounding region for a clipped region. If the resulting surrounding region consists of only very few pixels, then the clipped region is considered as a light source or a specularly reflected area. In this case, we enhance only the luma signal.

Fig. 6 shows an example of a surrounding region associated with the clipped region on the girl's arm. We observe that most unclipped pixels in the arm area (with similar chroma as the clipped region) that are close to the clipped region are selected as the surrounding region, which is used for the chroma estimation of the clipped pixels on the arm (Fig. 6c).

The C_b and C_r values of the clipped region could be interpolated from its surrounding region. A problem arises from the fact that the surrounding region is irregularly shaped with some "missing" pixels, which cannot be used in the interpolation because they are either clipped pixels or nonclipped pixels with different chroma to the current clipped region. A common interpolation approach is to use convolution (filtering). However, traditional convolution does not work when there are missing samples within the convolution mask.



Fig. 6. Example of a surrounding region. (a) Clipped image, (b) clipped areas (in color) superimposed on the image luma, and (c) the surrounding region (white pixels) for the clipped area on the girl's arm.

For the reason stated above, we use normalized convolution [32] instead, which allows for missing samples by adjusting the filter weights to use only the valid samples that fall within the convolution mask. The idea of normalized convolution is to associate with each pixel a certainty component m expressing the level of confidence in the pixel measurement. The certainty map m has the same dimension as the image.

To make the discussion more pertinent to our problem at hand, that is, interpolating chroma for saturated pixels, the normalized convolution can be expressed as follows:

$$\tilde{c}(x, y) = \frac{[c(x, y) \cdot m(x, y)] * h(x, y)}{m(x, y) * h(x, y)}, \quad (1)$$

where the certainty map $m(x, y)$ is 1 for the known pixels that are used in the interpolation, and $m(x, y)$ is 0 for the missing samples. The $c(x, y)$ and $\tilde{c}(x, y)$ represent the chroma channel signal (C_b or C_r) before and after the convolution, and $h(x, y)$ denotes the filter for performing the convolution. Here, a Gaussian filter with a standard deviation 5 is used as the function $h(x, y)$.

The normalized convolution mask $h(x, y)$ has a finite size. Consequently, a pixel located near the center of the clipped region may not have any pixel in the surrounding region lying in its mask. Hence, the pixel value cannot directly be corrected. In order to solve this problem, we choose to use the already corrected clipped pixels together with the surrounding region as the known data for estimating the uncorrected clipped pixels. In other words, the certainty map used is

$$m(x, y) = \begin{cases} 1, & \text{for unsaturated pixels in the surrounding region} \\ & \text{and saturated pixels that have been corrected,} \\ 0, & \text{otherwise.} \end{cases}$$

This helps improve the smoothness of the corrected chroma in the clipped region. Since already corrected pixels are used in the normalized convolution, the pixel order within a clipped region is very important. Because estimation error could propagate, the pixels with potentially less error should be corrected first.

In order to describe the smoothing process, we define a few notations here. Let Ω denote the saturated pixel set, that

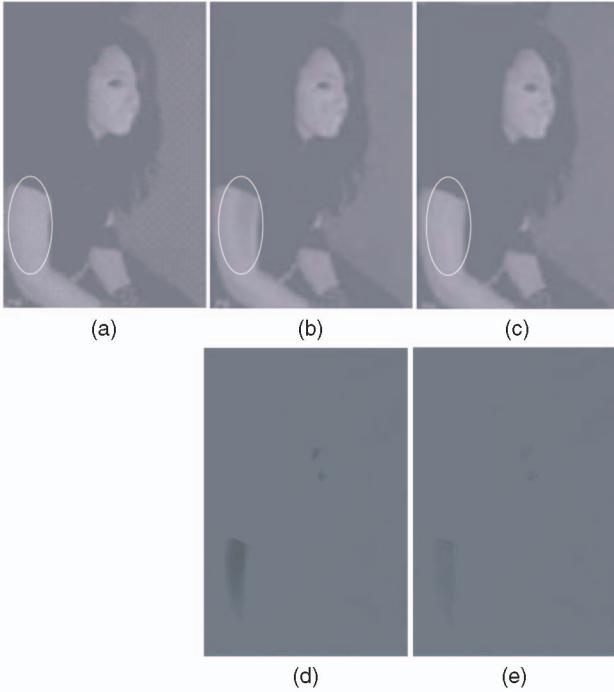


Fig. 7. Example of chroma correction. The Cr channel of (a) unclipped image, (b) clipped image, and (c) corrected image using our proposed algorithm. The difference between (a) and (b) is shown in (d), and the difference between (a) and (c) is shown in (e).

is, $\Omega = \{(x, y) : R(x, y) \geq \tau, \text{ or } G(x, y) \geq \tau, \text{ or } B(x, y) \geq \tau\}$. Furthermore, we use Ω_1 , Ω_2 , and Ω_3 to, respectively, represent the sets of clipped pixels with 1, 2, and 3 saturated channels.

Since the 1-channel saturated pixels Ω_1 tend to have less color distortion, and hence, less estimation error than 2- and 3-channel saturated pixels Ω_2 and Ω_3 , we first correct the pixels in Ω_1 , followed by pixels in Ω_2 , and finally, pixels in Ω_3 . Since clipped pixels that are close to the surrounding unclipped pixels, there is small estimation uncertainty, i.e., a small degree of error estimation for such pixels. For this reason, within each saturation category, we also sort the clipped pixels according to their distances to the nearest surrounding pixels, and first correct the ones closer to the surrounding region.

Fig. 7 shows the results of chroma correction using our method described in this section. Since the clipping in this image happens mostly in the red channel, we show the correction results by presenting the Cr channel (before and after correction) in Fig. 7. We can see in the circled area that the clipped image (Fig. 7b) is darker than the unclipped image (Fig. 7a), resulting in blocky distortion of the Cr channel. While the corrected chroma Cr , shown in Fig. 7c, is very close to that of the unclipped image. The above chroma correction result can be observed more easily in part Figs. 7d and 7e, where the difference between Fig. 7a and Fig. 7b, and the difference between Fig. 7a and Fig. 7c are shown.

2.4 Correct RGB Values

We calculate the missing R , G , or B values in each clipped region based on the estimated Cb and Cr values (calculated

in the previous step) and the unsaturated R , G , or B values in that region. We elaborate this correction process for the following three different scenarios, i.e., Ω_1 , Ω_2 , and Ω_3 .

2.4.1 Correct 2-Channel Saturated Pixels

Correction of 2-channel saturated pixels is the most straightforward scenario. We know that the conversion from RGB to YCbCr, introduced in the ITU-R BT.601 [33], is

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0.0627 \\ 0.5020 \\ 0.5020 \end{bmatrix}. \quad (2)$$

The above RGB and YCbCr values are within a range of 0.0-1.0. From (2), we have

$$Cb = [-0.1482 \ -0.2910 \ 0.4392] \times [R \ G \ B]^T + 0.5020, \quad (3)$$

$$Cr = [0.4392 \ -0.3678 \ -0.0714] \times [R \ G \ B]^T + 0.5020. \quad (4)$$

When two channels are clipped, then one of the R , G , and B values, say, U (which stands for the unsaturated channel), is known and the two clipped channels, say, S_1 and S_2 (which stand for the saturated channels), are unknown and need to be solved for. The U , S_1 , and S_2 are all components in the three color channels $[R, G, B]$. The corrected values of the two saturated channels can be uniquely solved for using the two equations (3) and (4). Therefore, we have

$$\begin{aligned} \tilde{S}_1 &= f_1(U, Cb, Cr), \\ \tilde{S}_2 &= f_2(U, Cb, Cr), \end{aligned}$$

where f_1 and f_2 are functions of U , Cb , and Cr , and \tilde{S} denotes the corrected value of color channel S . Note that we do not use Y to correct the RGB color channels, since Y is distorted when any color channel is clipped. The functions f_1 and f_2 can be derived uniquely from the RGB to CbCr conversion equations (3) and (4).

As an example, let us consider a case where R and G are the two clipped unknown channels in a saturated pixel, and B is the unclipped channel. From (3) and (4), we can solve for \tilde{R} and \tilde{G} given Cb , Cr , and B as follows:

$$\begin{aligned} \tilde{R} &= [(Cb - 0.5020 - 0.4392B) \times (-0.3678) \\ &\quad - (Cr - 0.5020 + 0.0714B) \times (-0.2910)] / \\ &\quad [(-0.1482) \times (-0.3678) - 0.4392 \times (-0.2910)], \\ \tilde{G} &= [(Cb - 0.5020 - 0.4392B) \times 0.4392 \\ &\quad - (Cr - 0.5020 + 0.0714B) \times (-0.1482)] / \\ &\quad [(-0.2910) \times 0.4392 - (-0.3678) \times (-0.1482)]. \end{aligned}$$

Any other 2-channel saturated pixels can be corrected in the same fashion.

2.4.2 Correct 1-Channel Saturated Pixels

Correction of 1-channel saturated pixels is similar to correcting 2-channel saturated pixels. Since there are only one unknown value S , and two equations (3) and (4), the value can be estimated twice by using the corrected Cb and Cr , respectively, as well as the two unsaturated channel

values U_1 and U_2 . Then, we simply take the average of the two estimations as the corrected value of the saturated channel. The estimation process can be described as

$$\begin{aligned}\tilde{S} &= \frac{S_1 + S_2}{2}, \text{ where} \\ S_1 &= f_3(U_1, U_2, Cb), \\ S_2 &= f_4(U_1, U_2, Cr),\end{aligned}$$

where the functions f_3 and f_4 are derived from (3) and (4), respectively.

As an example, let us consider a case where R is the clipped unknown channel, and G and B are the unclipped known channels. The corrected value \tilde{R} is computed as follows:

$$\begin{aligned}\tilde{R} &= \frac{R_1 + R_2}{2}, \text{ where} \\ R_1 &= \frac{Cb - (-0.2910) \times G - 0.4392 \times B - 0.5020}{-0.1482}, \\ R_2 &= \frac{Cr - (-0.3678) \times G - (-0.0714) \times B - 0.5020}{0.4392}.\end{aligned}$$

Any other channel saturated pixels (i.e., if G or B is clipped) can be corrected in the same fashion.

2.4.3 Correct 3-Channel Saturated Pixels

In the case of 3-channel saturated pixels, there are three unknown variables. Hence, three equations are needed to solve the corrected R , G , and B values. We first estimate the luma Y value of the 3-channel saturated pixels based on the surrounding region. We fit the clipped region and its surrounding area with a 2D Gaussian function. Unlike many other surface-fitting methods (e.g., [21]), we do not enforce any assumptions on the location or rotation of the 2D Gaussian function. By not assuming the center of the Gaussian function as the centroid of the clipped region, we are able to handle more general and sophisticated clipping cases. For example, our model works well for the situation where the brightest spot is not located near the center of the clipped region and the surrounding region only partially encloses the clipped area. In general, a 2D Gaussian function is of the following form:

$$g(x, y) = Ae^{-[a(x-x_0)^2 + 2b(x-x_0)(y-y_0) + c(y-y_0)^2]} + B,$$

where A , B , a , b , c , x_0 , and y_0 are the parameters, and $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$ is positive definite.

The least-squares surface-fitting problem can be solved using the following optimization form:

$$\begin{aligned}\operatorname{argmin}_{A, B, a, b, c, x_0, y_0} & \sum_{i=1}^n [Y_i - g(x_i, y_i)]^2, \\ \text{Subject to: } & \begin{bmatrix} a & b \\ b & c \end{bmatrix} \succ 0,\end{aligned}$$

where (x_i, y_i, Y_i) is the i th pixel in the surrounding area, x_i and y_i represent the pixel location, Y_i is the luma at pixel (x_i, y_i) , and the symbol " \succ " stands for positive definite. In order to remove the constraint from the above optimization problem, we apply variable substitutions. A symmetric and positive definite matrix M can be decomposed into

$M = LL^T$, where L is a lower triangular matrix [34]. For the matrix

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

in the constraint, we have

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \times \begin{bmatrix} l_{11} & l_{21} \\ 0 & l_{22} \end{bmatrix}.$$

Substituting a , b , and c using the new variables l_{11} , l_{21} , and l_{22} , the constraint is implied in the relation. Therefore, the optimization problem becomes unconstrained as follows:

$$\operatorname{argmin}_{A, B, l_{11}, l_{21}, l_{22}, x_0, y_0} \sum_{i=1}^n [Y_i - g(x_i, y_i)]^2,$$

where $g(x, y) = Ae^{-[\frac{l_{11}^2}{2}(x-x_0)^2 + l_{11}l_{21}(x-x_0)(y-y_0) + \frac{l_{21}^2 + l_{22}^2}{2}(y-y_0)^2]} + B$.

The optimization can be solved with a standard least-squares fitting algorithm. Once the parameters are estimated, the luma Y at the 3-channel saturated pixels can be computed by evaluating the Gaussian function. In the end, the corrected RGB values \tilde{R} , \tilde{G} , and \tilde{B} are solved using (2) as follows:

$$\begin{aligned}\begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} &= \begin{bmatrix} 1.16438356 & 0.00000030 & 1.59602688 \\ 1.16438356 & -0.39176253 & -0.81296829 \\ 1.16438356 & 2.01723263 & 0.00000305 \end{bmatrix} \\ &\times \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 0.0627 \\ 0.5020 \\ 0.5020 \end{bmatrix} \right).\end{aligned}$$

In the process of correcting saturated RGB values, we need to eliminate unrealistic corrected pixel values and obtain a stable enhancement algorithm. Hence, we set a lower bound α and an upper bound β as the multiplicative enhancement factor (i.e., the ratio between corrected value and clipped value) for each clipped pixel. We know the fact that the true values in the clipped channels should be greater than the clipped value. Therefore, the lower bound α is set to be greater than (or equal to) 1. To ensure a smooth transition between the unsaturated and saturated regions, the lower bound α is acted as a smooth enhancement factor mask to the saturated region. The mask can be denoted by:

$$\alpha = \begin{cases} (\alpha_0 - 1) \times \frac{d}{d_0} + 1, & \text{when } 0 < d < d_0, \\ \alpha_0, & \text{when } d \geq d_0, \end{cases}$$

where the enhancement factor α_0 is a constant and $\alpha_0 > 1$, the constant d_0 is the transition width, and d denotes the distance between the current saturated pixel and the closest unsaturated pixel in the surrounding region. The mask keeps the lower enhancement factor as α_0 for the pixels far from the unsaturated region, and gradually reduces the lower bound to 1 as the pixel gets closer to the unsaturated region.

2.5 Smooth Enhancement

The main purpose of color enhancement is to obtain visually plausible images and videos. Often, there are small jumps of enhanced values between adjacent 1, 2, and 3-channel saturated regions. This is because different strategies are

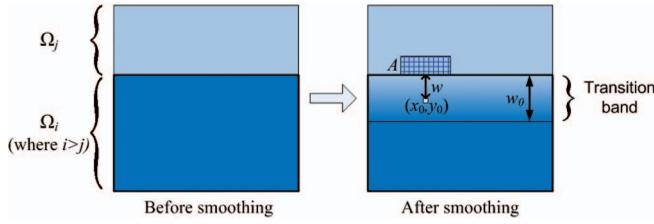


Fig. 8. Illustration of the smoothing process between 1, 2, and 3-channel saturated regions, i.e., Ω_1 , Ω_2 , and Ω_3 .

used for Ω_1 , Ω_2 , and Ω_3 when calculating saturated RGB channels from the corrected Cb and Cr , as described in Section 2.4. As a result, a smoothing process near the region boundaries of Ω_1 , Ω_2 , and Ω_3 is needed to reduce disturbing contours and obtain natural looking enhanced images.

In order to smooth the boundary between regions Ω_i and Ω_j , where $i > j$, among the pixels near the region boundary, we choose to adjust the pixels in Ω_i , where the pixels have more saturated channels and relatively higher estimation errors than those in Ω_j . Fig. 8 illustrates the smoothing process. We create a transition band with a width w_0 on the more saturated side (i.e., Ω_i) of the region boundary. The smoothed value at pixel (x_0, y_0) in the transition band is a linear combination of the estimated values (from the previous correction steps) at this pixel and its nearby region A_{x_0, y_0} . We define the area A associated with pixel (x_0, y_0) by first finding the pixel (x_1, y_1) that is closest to (x_0, y_0) and also in the less saturated region Ω_j . The area A_{x_0, y_0} is composed of (x_1, y_1) and its surrounding pixels in Ω_i that are within a distance of three pixels from (x_1, y_1) .

The adjusted saturated channel value $\tilde{P}(x_0, y_0)$ at (x_0, y_0) is given by

$$\begin{aligned} \tilde{P}(x_0, y_0) &= P(x_0, y_0) \times \frac{w}{w_0} + \left(1 - \frac{w}{w_0}\right) \\ &\times \frac{1}{N} \times \sum_{(x, y) \in A_{x_0, y_0}} P(x, y), \end{aligned} \quad (5)$$

where w_0 is the width of the transition band, w is the distance between (x_0, y_0) and (x_1, y_1) , and N is the number of pixels in the area A_{x_0, y_0} . The parameter w_0 can be chosen to adjust the amount of the smoothing. A reasonable range of w_0 is 3-10 pixels. The adjusted pixel value $\tilde{P}(x_0, y_0)$ is a linear combination of pixel values at (x_0, y_0) and A_{x_0, y_0} . The reason we take a small area A_{x_0, y_0} rather than a single pixel in Ω_j is to make the transition band smoother and avoid streaks in the band due to texture in Ω_j .

The effect of the smoothing process is illustrated in Fig. 9. Fig. 9a shows the saturation category map, with light gray, dark gray, and white representing Ω_1 , Ω_2 , and Ω_3 , respectively, and black being the unsaturated region. We observe that the enhanced image before the smoothing process has blocky artifacts between different saturated regions, while the enhanced image after smoothing appears more natural and visually pleasant.

3 EXPERIMENTAL RESULTS

In this section, we present some experimental results to show the effectiveness of our proposed method for

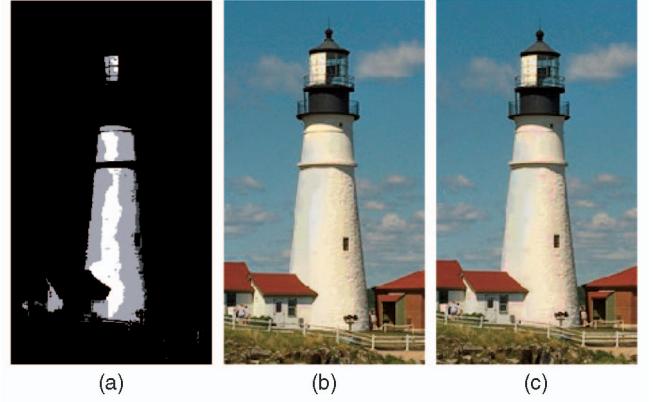


Fig. 9. Example of the smoothing effect. (a) The saturation category map, (b) enhanced image before smoothing, and (c) enhanced image after smoothing.

enhancing the clipped pixels. We use conventional 24 bits per pixel LDR color images for our tests. Thumbnails of our test images are shown in Fig. 10.

In our experiment, we generate the clipped images by clipping the R, G, B values that are greater than a threshold (e.g., 255×0.8 for 8 bits per color channel images). Then, we enhance the clipped images using our proposed method as well as the Bayesian algorithm [22], the only color correction algorithm for clipped pixels that we are aware of. To assess the quality of the corrected images, we compute the peak signal-to-noise ratio (PSNR) values (averaged over R, G , and B channels) of each test image for 1) the clipped image with no correction, 2) the enhanced image generated by the Bayesian algorithm, and 3) the enhanced image produced by our proposed algorithm. We also use two commonly used quality metrics, the CIELAB ΔE [35] and S-CIELAB [36], to evaluate the above-mentioned three situations. For each image, the ΔE



Fig. 10. Thumbnails of our test images. In reading order: girl, landscape, baby_girl, mountain, shoes, sunset, kodim03(caps), kodim05(motorcycles), kodim06(boat), kodim12(beach), kodim16(lake), kodim21(lighthouse), and kodim23(parrots).

TABLE 1
Quality Comparison between Different Algorithms

Image	PSNR (in dB)			CIELAB ΔE			S-CIELAB		
	Clipped	Bayesian	Ours	Clipped	Bayesian	Ours	Clipped	Bayesian	Ours
girl	42.23	39.70	50.68	6.68	9.39	2.51	0.98	1.51	0.33
landscape	25.66	28.69	29.53	11.79	9.27	8.39	1.87	1.41	1.29
baby_girl	32.15	29.06	36.40	8.40	11.39	4.92	1.24	1.86	0.68
mountain	29.98	34.27	37.02	11.52	5.87	4.18	1.65	0.91	0.53
shoes	25.34	32.93	32.44	13.03	6.32	5.66	2.00	1.02	0.77
sunset	21.73	20.76	27.18	18.62	16.96	10.06	3.04	3.47	1.72
kodim03 (caps)	34.34	35.24	39.71	12.37	12.25	6.47	2.30	2.25	1.07
kodim05 (motorcycles)	33.62	35.74	37.07	13.46	11.00	8.20	2.25	1.83	1.17
kodim06 (boat)	25.22	25.12	32.51	17.45	10.26	7.69	3.44	1.97	1.56
kodim12 (beach)	28.41	33.65	33.16	11.04	4.55	5.98	2.08	0.81	1.08
kodim16 (lake)	35.07	35.85	41.74	11.36	10.42	5.02	2.16	1.98	0.89
kodim21 (lighthouse)	32.40	33.56	36.76	16.43	13.35	8.46	3.03	2.51	1.49
kodim23 (parrots)	29.63	31.16	34.85	10.38	8.54	5.72	1.70	1.40	0.91
Average	30.44	31.98	36.08	12.50	9.97	6.40	2.14	1.76	1.04

or S-CIELAB metric is the averaged value over the saturated pixels in that image.

The image-quality comparison is listed in Table 1. From the table, we can see that while both the Bayesian algorithm and our proposed algorithm improve quality, our proposed method outperforms the Bayesian algorithm by an average of 4.10 dB in PSNR, 3.57 in CIELAB ΔE , and 0.72 in S-CIELAB over all test images. Our method performs well especially for images with large portion of clipped areas, such as baby_girl, sunset, and parrots images.

Fig. 11 shows the resulting images and is used for evaluating the subjective quality of the enhanced clipped pixels, and in turn, the overall image. For each image, we show (in reading order) the original image, clipped image, clipped areas superimposed on the image luma, enhanced image using the Bayesian algorithm, and enhanced image using our proposed algorithm. Pixel values of images in each group are linearly scaled using the same scaling factor to realize the maximum display contrast. In Fig. 11, we can see that all clipped images have color distortions due to overexposure. The Bayesian algorithm corrects color for most clipped regions. However, it overcorrects the color in some clipped regions and results in further color distortion. An overcorrection example can be seen in the background area of the “baby_girl” image. These artifacts happen when the color properties of the clipped region are different from the statistical properties of the unclipped regions in the image. Distortion usually occurs when the images do not possess much color variety or a large portion of clipped pixels exists. Compared to the Bayesian algorithm, our algorithm gives comparable or better subjective quality, without notable artifacts.

Our method works well when a saturated region is associated with an unsaturated surrounding region with similar chroma. In some cases, no such surrounding region can be found, and our method cannot be used to estimate the chroma in the clipped region. These cases are extremely difficult to handle due to the lack of useful information. A possible solution is to use a classifier, as developed in [17], to classify these clipped regions as lights, reflections, or diffuse surfaces. Then, the brightness of each class of objects is enhanced by a multiplicative factor. The classifier, however, usually requires human interaction. Furthermore, the multiplicative factors are set to rather arbitrary values (1.5 for lights, 1.25 for reflections).

One important application of the clipped pixel color enhancement is to use it as a preprocessing step of an inverse tone mapping for producing high-quality HDR images/video from existing LDR images/video. Since color clipping is often more perceptible in high-contrast inverse-tone-mapped HDR images, correcting the clipping appears more important for generating and displaying HDR images. In order to verify the importance of color correction for the clipped pixels in the contrast enhancement process, we apply an inverse tone mapping to convert LDR images into HDR images. Since a logarithmic function is the empirical model of the tone mapping operators, we use the inverse of the logarithmic function as the inverse tone mapping operator to expand the contrast of the LDR images. The subjective quality of the inverse-tone-mapped HDR images cannot be directly shown on a conventional computer screen or print, due to the limited dynamic range of such media. For this reason, we present two “virtual” exposures of each HDR image, as done in [16], [21], to display HDR images in print.

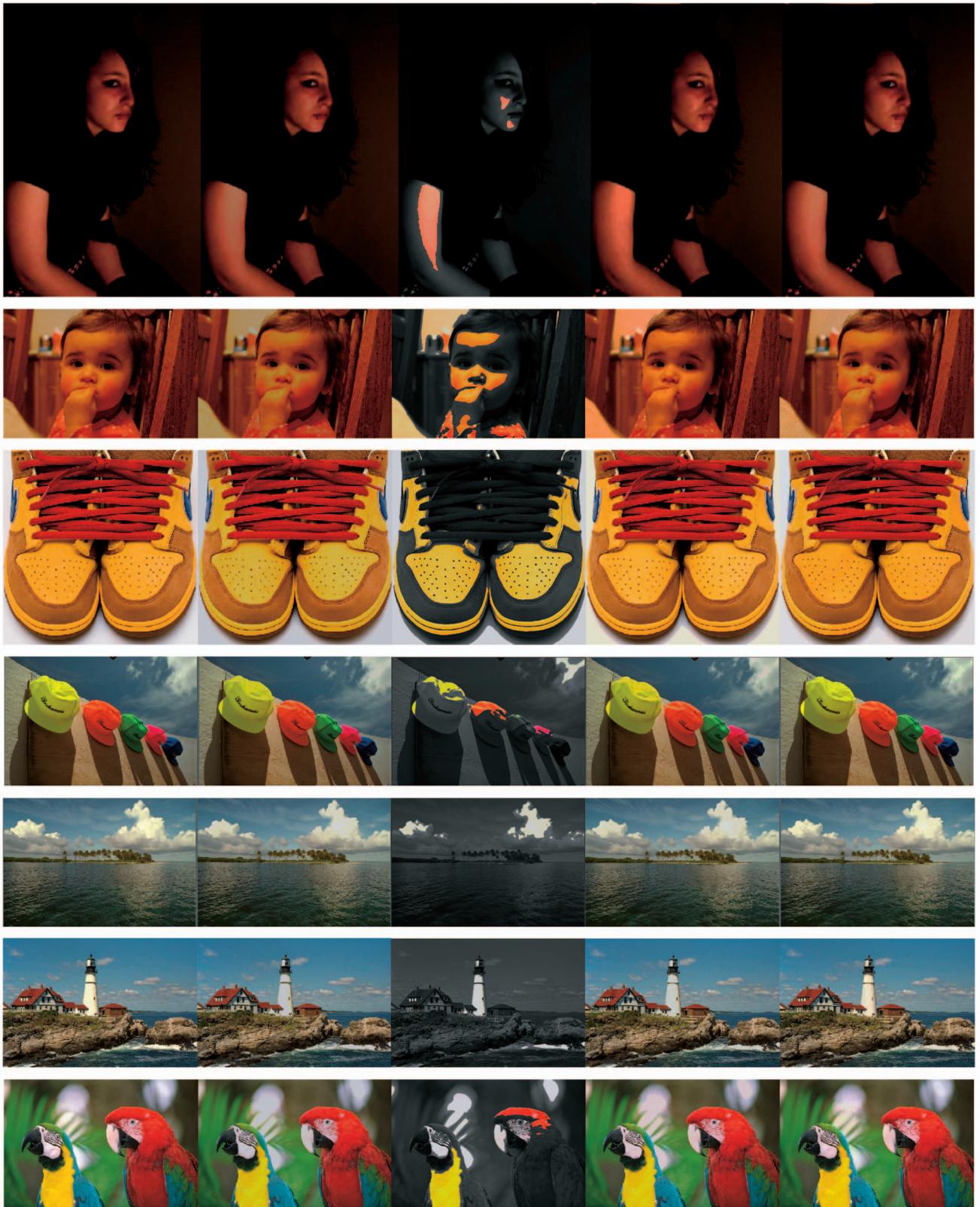


Fig. 11. Results of clipped pixel enhancement. For each row, we show (from left to right) the original image, clipped image, clipped areas superimposed on the image luma, enhanced image using Bayesian algorithm, and enhanced image using our proposed algorithm.

Each exposure reveals different brightness ranges of the entire dynamic range of the HDR image. Fig. 12 shows two virtual exposures of the HDR “girl” and “baby_girl” image sets. Each column of images corresponds to (from left to

right) the original unclipped image, the clipped image, the enhanced image produced by the Bayesian algorithm, and the enhanced image obtained by our proposed algorithm, respectively. In Fig. 12, especially the low exposure images,

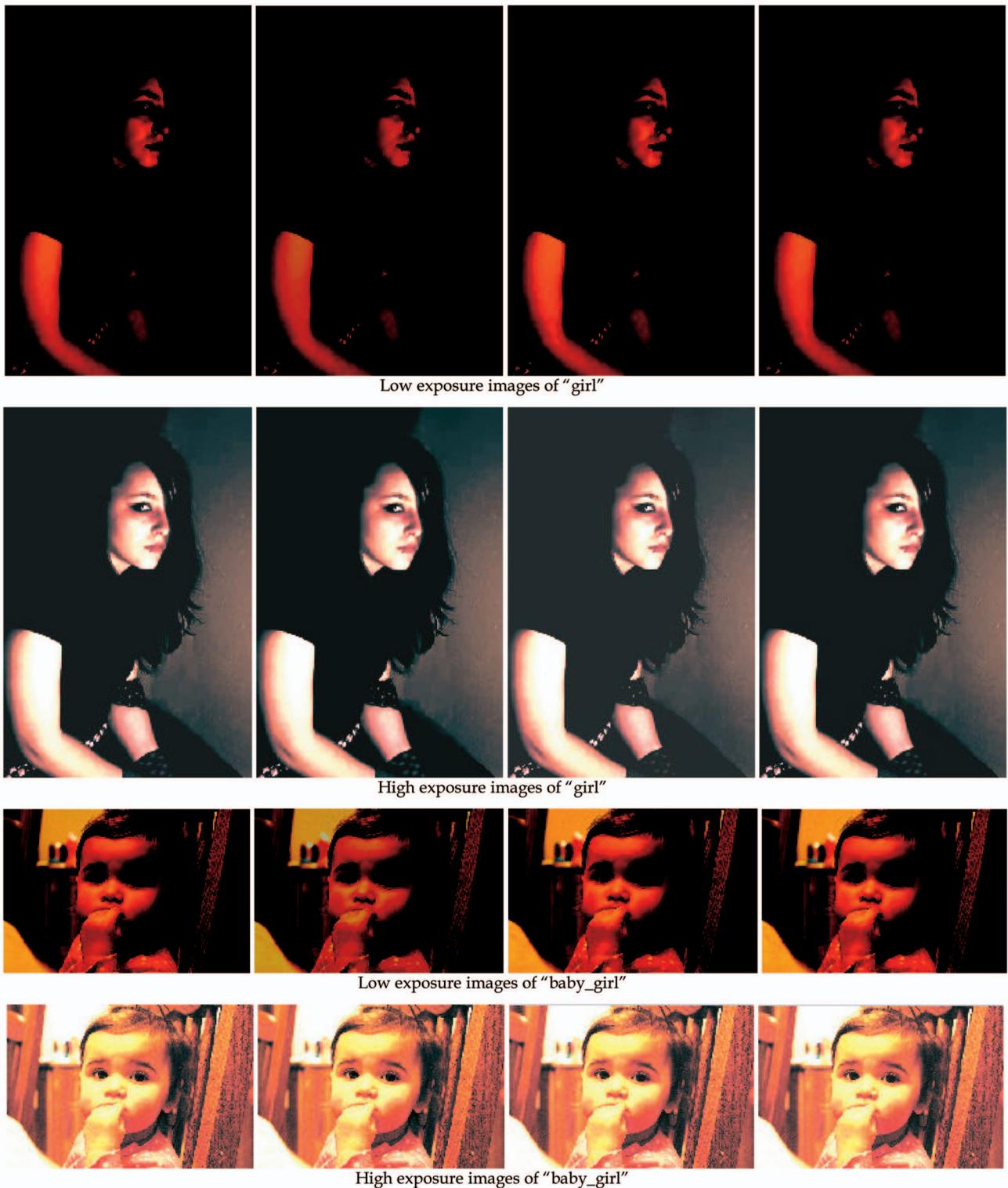


Fig. 12. Two virtual exposures of HDR images. The four columns of images correspond to (from left to right) the original unclipped LDR image, the clipped LDR image, the enhanced image obtained by the Bayesian algorithm, and the enhanced image produced by our proposed algorithm, respectively.

we observe that the image corrected with our approach is more similar to the original than either the clipped image or the result of Bayesian algorithm. This verifies that our color enhancement algorithm produces high-quality HDR images from clipped LDR images.

4 CONCLUSIONS

In this paper, we have proposed an effective method for enhancing clipped pixels in color images. We take advantage of the strong correlation between the chroma of the clipped pixels and their surrounding unclipped pixels. Our

method greatly reduces the color distortion caused by clipping. It also effectively corrects the luma of pixels in the clipped areas. Our proposed method outperforms the Bayesian algorithm by an average of 4.10 dB in PSNR, 3.57 in CIELAB ΔE , and 0.72 in S-CIELAB. Subjective results also show that the enhanced images generated by our method are visually more plausible than the clipped images and the enhanced images produced by the Bayesian algorithm.

Many other research areas can benefit from the color correction for clipped pixels. We showed that by applying inverse tone mapping to LDR images that have been enhanced by our method, we obtain more plausible and realistic HDR images than applying inverse tone mapping directly to the clipped images. Since color has been widely used in machine-based vision systems, our algorithm may also help increase the performance of tasks such as color-based image segmentation, object recognition, tracking, panoramic images generating, and multiview image processing.

ACKNOWLEDGMENTS

This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

- [1] J.A. Ferwerda, "Elements of Early Vision for Computer Graphics," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 22-33, Sept. 2001, doi:10.1109/38.946628.
- [2] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2006.
- [3] K. Myszkowski, R. Mantiuk, and G. Krawczyk, *High Dynamic Range Video*. Morgan & Claypool Publishers, 2008, doi:10.2200/S00109ED1V01Y200806CGR005.
- [4] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive Logarithmic Mapping for Displaying High Contrast Scenes," *Computer Graphics Forum*, vol. 24, no. 3, pp. 419-426, 2003, doi:10.1111/1467-8659.00689.
- [5] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic Tone Reproduction for Digital Images," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 267-276, 2002, doi:10.1145/566654.566575.
- [6] E. Reinhard and K. Devlin, "Dynamic Range Reduction Inspired by Photoreceptor Physiology," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 1, pp. 13-24, Jan. 2005, doi:10.1109/TVCG.2005.9.
- [7] G. Ward, H. Rushmeier, and C. Piatko, "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes," *IEEE Trans. Visualization and Computer Graphics*, vol. 3, no. 4, pp. 291-306, Oct. 1997, doi:10.1109/2945.646233.
- [8] G. Krawczyk, K. Myszkowski, and H.-P. Seidel, "Lightness Perception in Tone Reproduction for High Dynamic Range Images," *Computer Graphics Forum*, vol. 24, no. 3, pp. 635-645, 2005, doi:10.1111/j.1467-8659.2005.00888.x.
- [9] R. Fattal, D. Lischinski, and M. Werman, "Gradient Domain High Dynamic Range Compression," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 249-256, 2002, doi:10.1145/566570.566573.
- [10] R. Mantiuk, K. Myszkowski, and H.-P. Seidel, "A Perceptual Framework for Contrast Processing of High Dynamic Range Images," *ACM Trans. Applied Perception*, vol. 3, no. 3, pp. 286-308, 2006, doi:10.1145/1166087.1166095.
- [11] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs, "High Dynamic Range Display Systems," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 760-768, Aug. 2004, doi:10.1145/1015706.1015797.
- [12] O.A. Akyüz, R. Fleming, B.E. Riecke, E. Reinhard, and H.H. Bühlhoff, "Do HDR Displays Support LDR Content? A Psychophysical Evaluation," *ACM Trans. Graphics*, vol. 26, no. 3, July 2007, doi:10.1145/1276377.1276425.
- [13] L. Meylan, S. Daly, and S. Süsstrunk, "The Reproduction of Specular Highlights on High Dynamic Range Displays," *Proc. 14th Color Imaging Conf.*, oai:infoscience.epfl.ch:85838, 2006.
- [14] L. Meylan, S. Daly, and S. Süsstrunk, "Tone Mapping for High Dynamic Range Displays," *Proc. Conf. IS&T/SPIE Electronic Imaging: Human Vision and Electronic Imaging XII*, 2007, doi:10.1117/12.706472.
- [15] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Inverse Tone Mapping," *Proc. Fourth Int'l Conf. Computer Graphics and Interactive Techniques in Australasia and Southeast Asia (GRAPHITE '06)*, pp. 349-356, Nov./Dec. 2006, doi:10.1145/1174429.1174489.
- [16] A.G. Rempel, M. Trentacoste, H. Seetzen, H.D. Young, W. Heidrich, L. Whitehead, and G. Ward, "LDR2HDR: On-the-Fly Reverse Tone Mapping of Legacy Video and Photographs," *ACM Trans. Graphics*, vol. 26, no. 3, 2007, doi:10.1145/1276377.1276426.
- [17] P. Didyk, R. Mantiuk, M. Hein, and H.-P. Seidel, "Enhancement of Bright Video Features for HDR Displays," *Computer Graphics Forum*, vol. 27, no. 4, pp. 1265-1274, 2008, doi:10.1111/j.1467-8659.2008.01265.x.
- [18] G.J. Klinker, S.A. Shafer, and T. Kanade, "The Measurement of Highlights in Color Images," *Int'l J. Computer Vision*, vol. 2, pp. 7-32, 1988.
- [19] H.-L. Shen and Q.-Y. Cai, "Simple and Efficient Method for Specularity Removal in an Image," *Applied Optics*, vol. 48, no. 14, pp. 2711-2719, 2009, doi:10.1364/AO.48.002711.
- [20] P. Tan, S. Lin, L. Quan, and H.-Y. Shum, "Highlight Removal by Illumination-Constrained Inpainting," *Proc. Ninth IEEE Int'l Conf. Computer Vision*, pp. 164-169, 2003, doi:10.1109/ICCV.2003.1238333.
- [21] L. Wang, L.-Y. Wei, K. Zhou, B. Guo, and H.-Y. Shum, "High Dynamic Range Image Hallucination," *Proc. ACM SIGGRAPH*, no. 72, 2007, doi:10.1145/1278780.1278867.
- [22] X. Zhang and D.H. Brainard, "Estimation of Saturated Pixel Values in Digital Color Imaging," *J. Optical Soc. Am. A*, vol. 21, no. 12, pp. 2301-2310, 2004, doi:10.1364/JOSAA.21.002301.
- [23] B.K. Gunturk, J. Glotzbach, Y. Altunbasak, R.M. Mersereau, and R.W. Schafer, "Demosaicking: Color Filter Array Interpolation," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 44-54, Jan. 2005, doi:10.1109/MSP.2005.1407714.
- [24] S. Farsiu, M. Elad, and P. Milanfar, "Multiframe Demosaicking and Super-Resolution of Color Images," *IEEE Trans. Image Processing*, vol. 15, no. 1, pp. 141-159, Jan. 2006, doi:10.1109/TIP.2005.860336.
- [25] J.E. Adams and J.F. Hamilton, Jr., "Adaptive Color Plane Interpolation in Single Color Electronic Camera," US patent 5 506 619, Apr. 1996.
- [26] Kodak Lossless True Color Image Suite, <http://r0k.us/graphics/kodak/>, 2010.
- [27] H. Landis, "Production-Ready Global Illumination," *SIGGRAPH Course Notes 16*, 2002.
- [28] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. IEEE Int'l Conf. Computer Vision*, 1998, doi:10.1109/ICCV.1998.710815.
- [29] F. Durand and J. Dorsey, "Fast Bilateral Filtering for the Display of High-Dynamic-Range Images," *ACM Trans. Graphics (TOG)*, vol. 21, no. 3, July 2002, doi:10.1145/566570.566574.
- [30] S. Paris and F. Durand, "A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach," *Proc. European Conf. Computer Vision*, 2006, doi:10.1007/s11263-007-0110-8.
- [31] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*. Addison-Wesley Longman Publishing Co., 1992.
- [32] H. Knutsson and C.-F. Westin, "Normalized and Differential Convolution: Methods for Interpolation and Filtering of Incomplete and Uncertain Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 515-523, June 1993.
- [33] ITU-R, Rec. BT. 601-4, *Encoding Parameters of Digital Television for Studios*.
- [34] J.E. Gentle, "Cholesky Factorization," *Numerical Linear Algebra for Applications in Statistics*, pp. 93-95, Springer-Verlag, 1998.
- [35] CIE (Commission Internationale de l'Éclairage). Colorimetry technical report. CIE Pub. No. 15, Bureau Central de la CIE [corrected reprint 1996], 1986.
- [36] X. Zhang and B.A. Wandell, "A Spatial Extension of CIELAB for Digital Color-Image Reproduction," *J. Soc. Information Display*, vol. 5, no. 1, pp. 61-63, Mar. 1997, doi:10.1889/1.1985127.



Di Xu (S'06) received the BS degree in electrical engineering from Beijing Normal University, China, in 2003, and the MASc degree in electrical and computer engineering from the University of Victoria, British Columbia, Canada, in 2007. She is currently working toward the PhD degree in the Department of Electrical and Computer Engineering at the University of British Columbia, Vancouver, Canada. Her research interests are focused on the fields of digital image and video processing. These include high dynamic range image and video generation, video quality metric, digital video transcoding, and image coding. She is a student member of the IEEE.



Colin Doutre (S'06) received the BSc degree in electrical engineering from Queen's University, Kingston, Ontario, Canada, in 2005, and the MASc degree in electrical and computer engineering from the University of British Columbia (UBC), Vancouver, Canada, in 2007, where he is currently working toward the PhD degree. His research interests include multiview video processing, 3D video, and video compression. He has received numerous awards and scholarships,

including the Professional Engineers of Ontario Gold Medal for being the top-ranked engineering student graduating from Queen's University in 2005 and the Governor General's Gold Medal for being the top research-based master's student graduating from UBC in 2007. He is a student member of the IEEE.



Panos Nasiopoulos (M'96) received the bachelor's degree in physics from the Aristotle University of Thessaloniki, Greece, and the bachelor's, master's, and PhD degrees in electrical and computer engineering from the University of British Columbia, Canada. He is a professor in the Department of Electrical and Computer Engineering at the University of British Columbia (UBC), the holder of the Dolby professorship in digital multimedia, and the

current director of the Master of Software Systems Program at UBC. Before joining the UBC, he was the president of Daikin Comtec US and executive vice president of Sonic Solutions. He is a member of the IEEE, the ACM, the Standards Council of Canada (International Standards Organization/International Telecommunications Union (ISO/ITU)), and the Moving Pictures Experts Group (MPEG).

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**