SMPTE Meeting Presentation

Automatic, Fast and Perceptually Accurate Gamut Mapping Based on Vision Science Models

Syed Waqas Zamir

Universitat Pompeu Fabra, Barcelona, Spain, waqas.zamir@upf.edu

Javier Vazquez-Corral

Universitat Pompeu Fabra, Barcelona, Spain, javier.vazquez@upf.edu

Marcelo Bertalmío

Universitat Pompeu Fabra, Barcelona, Spain, marcelo.bertalmio@upf.edu

Written for presentation at the
SMPTE 2017 Annual Technical Conference & Exhibition

Abstract. *Gamut mapping transforms colors of the original (image or video) content to the color palette of the display device with the simultaneous goals of (a) reproducing content accurately while preserving the artistic intent of the original content's creator and (b) exploiting the full color rendering potential of the target display device. The rapid advancement in display technologies has created a pressing need to develop automatic and fast gamut mapping algorithms that can deal with imagery intended for both conventional and emerging displays. In this paper, we propose a novel framework based on retinal and color perception models from vision science that offers a functionality to perform both gamut reduction and gamut extension, while preserving hue and taking into account the analysis of the colors of the input image. We evaluate the performance of the proposed framework visually and by using a perceptually-based error metric, according to which the gamut-mapped results of our framework outperform those of the state-of-the-art methods.*

Keywords. Gamut mapping, wide color gamut, color reproduction, gamut mapping algorithms.

Introduction

For the past several decades, the primary focus of both cinema and television (TV) industries is in the improvement of image/video quality features such as spatial resolution, temporal resolution (frame rate), contrast and colors. However, it is only due to the recent technological breakthroughs, that manufacturers are able to introduce to the marketplace new flat-panel TVs and digital cinema projectors which are capable of generating high contrast and reproducing wide range of colors; thus providing users a visual experience closer to reality.

Owing to the trichromacy property of human vision, virtually all display devices work in the following manner: they mix three well chosen red, green and blue color primaries in different proportions to generate different colors. The range of colors that a device is able to reproduce is called its color gamut. Although a 3D color volume is a more complete way to describe the actual color palette that a device can reproduce, for the sake of simplicity it is a common practice to decouple luminance from the chromatic content referred as chromaticity and show only the chromatic content on a 2D plane. Figure 1 shows the standard CIE xy chromaticity diagram, where the horseshoe-shaped region corresponds to the chromaticities of all the colors that a standard observer can perceive. Practically all commercially available displays have a triangle-shaped gamut in the CIE xy chromaticity diagram due to the use of three color primaries. Therefore, they are unable to reproduce many colors that we can see.

Different display devices can have different gamuts depending on the set of color primaries they use. This posses a serious problem as we cannot create image content differently for all these different display devices. To tackle this challenge, there exist some standard color gamuts: DCI-P3 gamut is used in cinema industry and BT.709 gamut is for HDTVs, broadcasts, online streaming, etc.; both of these gamuts are shown in Figure 1. Displays receive input signals with standard color gamuts, but each display has to "fit" the gamut of the incoming content to its native color gamut by using an automated procedure called gamut mapping (GM). GM transforms colors of a source material to the colors of a target device. Generally there exist two possibilities for gamut mapping[1]: if the source gamut is larger than the target gamut, gamut reduction (GR) is needed[2-8] and if the source gamut is smaller than the target gamut, the process of gamut extension (GE) is desired[9-13].

Gamut mapping is also needed on the image capture side. There exists various consumer and professional movie cameras that are capable of capturing content with different range of colors. Although high-end cameras provide an option to save content in RAW format and process it offline, most cameras have to yield content with standard gamuts on the fly. Reproducing content offline is not even feasible in some cases such as in live TV broadcasts and in low budget movie productions, hence GM is carried out within the camera.

In the cinema industry, colorists at the postproduction stage make color and contrast adjustments in the movie to improve storytelling. Once the artistic signature is added, the colorist has to create two movie grades for distribution: one with the DCI-P3 gamut for cinema viewing and the other with the BT.709 gamut for HDTV viewing. Now with the introduction of very wide BT.2020 standard (shown in Figure 1), it is foreseeable that in the near future colorists will have to create
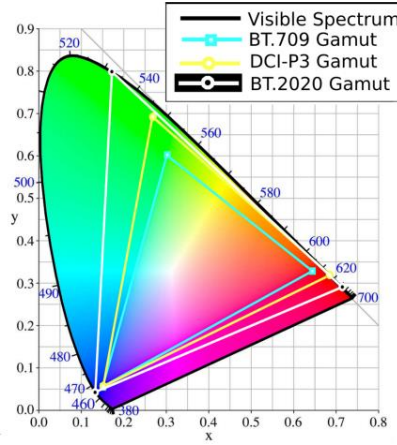
Figure 1. Standard gamuts on CIE xy chromaticity diagram.

at least three movie grades. Therefore, it makes gamut mapping very desirable in the cinema industry too.

Perceptually-based Gamut Mapping Framework

In the CIE xy chromaticity diagram, the least saturated colors are near x = y = 0.3 and as we move away from this point, colors become more and more saturated. In HSV (H for hue, S for saturation and V for value) color space, the goal of gamut reduction can be posed as reducing saturation and the goal of gamut extension is related to increasing saturation. (The method to transform RGB images into HSV color space can be found in Appendix.) Following the works[14,15] that are based on retinal and color perception models, in this paper we express gamut mapping as a convolution of a kernel (K) with the saturation component (S_0) of the input image plus a constant-valued image (P):

$$S_i = S_0 * K + P \qquad (1)$$

where S_i is an interim saturation image and the symbol $*$ denotes the classical convolution operation. All the pixel values of the constant-valued image P are set to the product of the mean value of the input saturation channel S_0 and the mean value of the kernel (K). Meanwhile the kernel K is computed as:

$$K = \mathcal{F}^{-1}\left(\frac{1}{(\alpha + \beta - \gamma) + \gamma\mathcal{F}(\omega)}\right) \qquad (2)$$

where $\mathcal{F}$ is a Fourier transform operation, ω is a Gaussian kernel with standard deviation σ, and α and β are positive constants. γ is a positive constant in the case of gamut extension and it is negative for gamut reduction. For the purpose of illustration, we show the shape of kernel K for
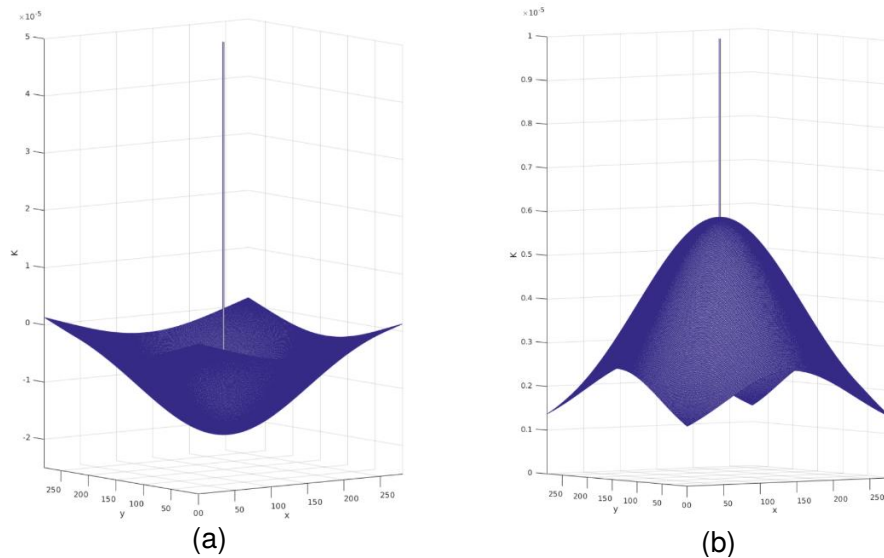
Figure 2. Examples of kernel (K): (a) for gamut extension, i.e. when $\gamma > 0$, and (b) for gamut reduction, i.e. when $\gamma < 0$.

gamut extension and for gamut reduction in Figure 2a and Figure 2b, respectively. It can be noticed that in the case of gamut extension the kernel K has a tendency to increase the contrast of the input image, whereas in the case of gamut reduction it averages the image values, therefore decreases the contrast.

In gamut extension the goal is to increase image saturation (using $\gamma > 0$); since our method works by increasing contrast, it may end up decreasing the saturation of some image pixel values. Similarly, the goal of gamut reduction is to decrease image saturation (using $\gamma < 0$); since in the case of gamut reduction our method works by decreasing contrast, it may end up increasing the saturation of some pixels of the input image. Both of these aforementioned behaviors are undesirable, therefore once we have the interim image S_i, we apply to it a threshold operation which is $S_f = \max(S_0, S_i)$ in the case of gamut extension (to avoid pixels from getting desaturated), and $S_f = \min(S_0, S_i)$ in the case of gamut reduction (to avoid pixels from getting saturated).

It is important to mention that the aforementioned procedure is sufficient to perform gamut mapping: both gamut reduction and gamut extension. However, it can be helpful in improving results for gamut reduction and gamut extension if the following particular (but optional) operations are used. (*Let us note that all the results provided in this paper are also obtained by using the following added steps.*)
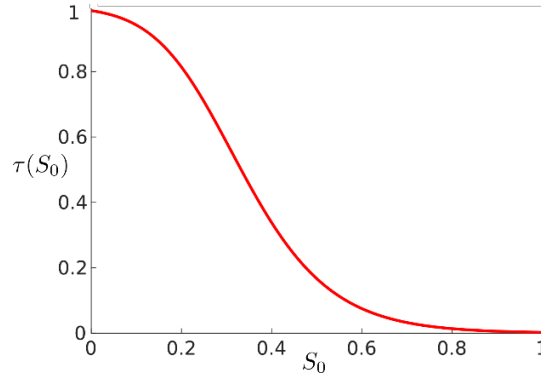
Figure 3. Weighting function $\tau(\ )$ to give weights to each pixel of the input image.

Natural-Tones Preserving Gamut Extension

In gamut extension, certain colors require a special treatment such as skin tones, less saturated natural objects, neutral colors and some particular memory colors. To incorporate such functionality in the main gamut mapping framework, the saturation image (S_f) obtained after threshold operation is linearly combined with the saturation channel of the input image (S_0) as:

$$S_{final} \ = \ \tau(S_0)S_0 + (1 - \tau(S_0))S_f \tag{3}$$

where the weighting function $\tau(\ )$ takes as an argument the saturation of the input image (S_0) and associates weights with the pixels of the image in order to treat low-saturated and high-saturated pixels differently (as shown in Figure 3), and the function $\tau(\ )$ is computed as:

$$\tau(S_0(x)) = \ 1 - \frac{1}{\left(1 + 0.55e^{-1.74S_0(x)}\right)^2} \tag{4}$$

Where x is the pixel location.

Iterative Gamut Reduction

When the value of γ parameter is negative for the Kernel (K) in Eq. (2), our gamut mapping framework behaves as a gamut reduction procedure and yields reduced-gamut output images. However, even better results can be obtained by making the same procedure work in an iterative manner as follows. At iteration level one, we run Eq. (1) by using $\alpha = 0$, $\beta = 1$ and $\gamma = 0$ and check if there are some pixels that lie inside the destination gamut. If yes, we mark these pixels as a part of the final reduced-gamut image and these values will not be modified in subsequent

Table 1. Primaries of gamuts.

Gamuts	Red Primaries		Green Primaries		Blue Primaries	
	x	y	x	y	x	y
BT.2020	0.708	0.292	0.170	0.797	0.131	0.046
DCI-P3	0.680	0.320	0.265	0.690	0.150	0.060
BT.709/sRGB	0.640	0.330	0.300	0.600	0.150	0.060
Toy	0.570	0.320	0.300	0.530	0.190	0.130

iterations. We move to the next iteration and perform gamut reduction procedure again but now with a kernel (K) where the value of γ has been decreased (for instance, the decrement in γ value can be 0.05, and setting $\alpha = \frac{|\gamma|}{20}$). Again we find those pixels that were out-of-gamut in the previous iteration but now moved inside the destination gamut, we make these pixels a part of the final image and would not modify them in the following iterations. This iterative process continues until all the out-of-gamut colors are mapped inside the destination gamut.

Experiment and Results

In this section we present results both for gamut extension and gamut reduction. We first convert the input RGB image into HSV color space and then apply our method only on the saturation component, while keeping both hue and value components constant.

Evaluation of Gamut Extension Algorithms (GEAs)

To extend the color gamut of an image using Eq. (3), we have used the following set of parameters: $\beta = 1$, $\gamma = 0.25$, $\alpha = \frac{|\gamma|}{20}$, and the standard deviation σ for Gaussian kernel ω is set to one-third of the number of rows or columns of the input image (whichever is larger). In order to compare the performance of different GEAs, we perform mapping from a smaller 'Toy' gamut to a larger BT.709 gamut. Color primaries for both of these gamuts are mentioned in Table 1. We compute results using our GEA and the following four methods of Laird et al.[12]:

- **Same drive signal (SDS)** method linearly maps the RGB primaries of the source gamut to the RGB primaries of the destination device gamut;
- **Hybrid color mapping (HCM)** algorithm combines the input image with the output of SDS method;
- **Chroma extension** method extends the chroma of the input image while keeping both hue and lightness constant;
- **Lightness chroma adaptive (LCA)** algorithm performs extension by modifying both chroma and lightness of the input image without changing the hue.

Figure 4 presents the results of competing GEAs, where it can be seen that the images produced by our GEA are perceptually more faithful to the original images (first column) than the other

Figure 4. Reproductions of GEAs. Column 1: input images. Column 2: HCM. Column 3: SDS Column 4: Chroma extension. Column 5: LCA. Column 6: Our GEA. Original images are from the datasets[16,17].

methods[12]. Particularly, notice in row 1 that the reproduction of skin tones (which is always a key issue in movie postproduction) is more accurate in the case of our GEA (sixth column) than other competing GEAs. For instance, the SDS method performs excessive color extension and therefore ends up rendering flesh tones poorly (see third column); similar tendency can be observed in the results of chroma extension method (fourth column) and LCA algorithm (fifth column).

We present in Figure 5 some close-ups (taken from Figure 4) for a better comparison. The LCA algorithm modifies chroma as well as lightness of the input image and yields an output image with enhanced contrast. This enhancement of contrast may produce artifacts and cause some pixels to go towards black (loss of saturation and spatial detail); as an example, see the lamp and alphabet in row 1 of Figure 5. Rows 3 and 4 of the same figure show that the chroma extension method gives an excessive boost to the chroma of the input image and produces an over-saturated result. One can see in row 4 of Figure 5 that our method applies a small amount of extension to low-saturated objects, while extending high saturated colors normally. This is a very useful property to preserve colors of natural objects and providing a special treatment to skin tones, neutral colors and some particular memory colors. Since SDS algorithm contributes in the result of HCM algorithm, the HCM algorithm may produce over-saturated images if the contribution from SDS method is high; for instance, see row 4 and second column of Figure 5.

The gamut extension results presented in this paper are obtained using a fixed value of γ parameter in Eq. (2), which implies that in some cases the proposed GEA may not provide optimal images: they may be under-enhanced if the value for γ is too small or over-enhanced if the γ value is too large, see for example Figure 6. Therefore there is a need to devise a method for making adjustment in γ value by taking into account the content of input image and the difference between the source and target gamuts, and we leave this as a future work.
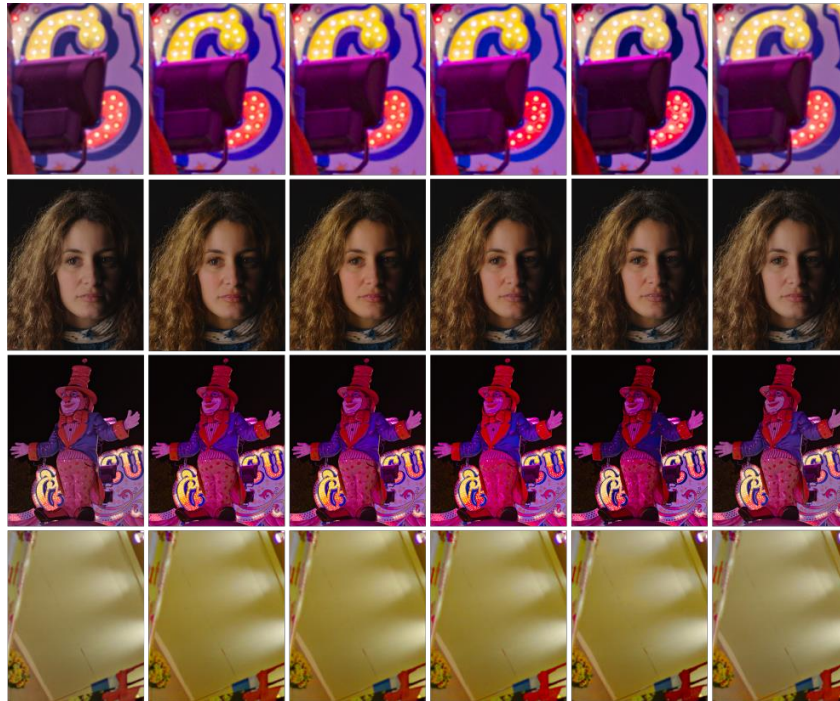
Figure 5. Comparison of GEAs: crops are from Figure 4. Column 1: input images. Column 2: HCM. Column 3: SDS Column 4: Chroma extension. Column 5: LCA. Column 6: Our GEA.

Figure 6. Effect of γ parameter on gamut extension. Left: ground truth image. Middle: under-enhanced result when γ value is too small ($\gamma = 0.1$). Right: over-enhanced result when γ value is too large ($\gamma = 0.5$). Note that these specific γ values produce under and over enhanced results only for this particular image but they may yield perceptually faithful results for other images.

Evaluation of Gamut Reduction Algorithms (GRAs)

To obtain the reduced-gamut images, we map the colors of BT.709/sRGB images to a smaller 'Toy' gamut. We apply the proposed GRA on the saturation channel of the input images by using the following parameter values in Eq. (1): $\beta = 1$, $\alpha = \frac{|\gamma|}{20}$, and the standard deviation σ for Gaussian kernel ω is equal to one-third of the number of rows or columns of the input image

(whichever is larger). It is important to recall that our GRA is iterative in nature, and between each iteration we decrease the gamma value by making a change of $\Delta\gamma = 0.05$. For the purpose of evaluation, results are computed by using the proposed GRA and the following four methods:

- **LCLIP** method[3] clips the chroma of out-of-gamut colors to the destination gamut boundary along lines of constant hue and lightness;
- **POSTCLIP** algorithm[18] first uses a 3 x 3 linear matrix to transform colors of the input image to the color gamut of the destination device. Then it clips those values that fall outside the extremes of the coding range;
- **HPMINDE** (Hue Preserving Minimum ΔE) method[2] involves clipping of the out-of-gamut color to the closest color, in terms of ΔE error, on the boundary of destination gamut along lines of constant hue;
- **Schweiger et al.**[4] make use of a compression function that squeezes colors near the destination gamut boundary in order to accommodate the out-of-gamut colors.

Figure 7 shows some visual examples of image reproduction quality of some GRAs, where it can be observed that the images produced by our GRA are perceptually more faithful to the original images (first column) than those of the other methods. It is noticeable in the same figure that the proposed GRA is capable of preserving hues and retaining spatial detail better than the other competing GRAs. Particularly, notice in the close-ups shown in Figure 8 that HPMINDE[2] (fourth column) introduces artifacts in the reproduced images because it may project two nearby out-of-
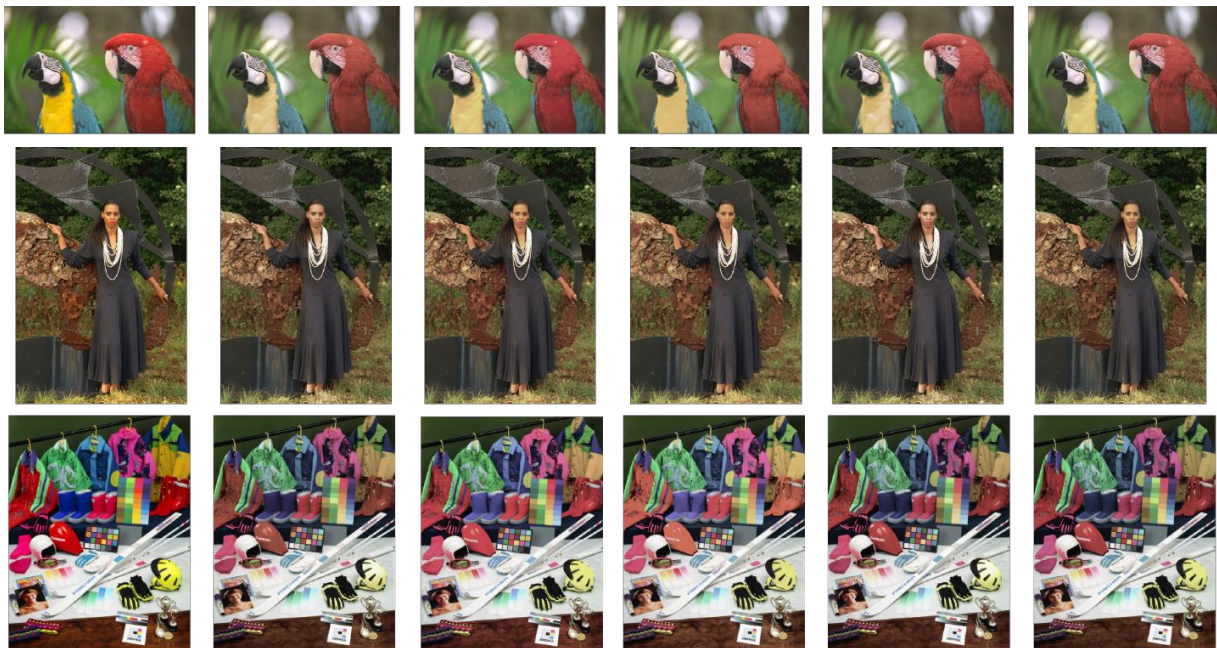

Figure 7. Reproductions of GRAs. Column 1: input images. Column 2: LCLIP[3]. Column 3: POSTCLIP[18]. Column 4: HPMINDE[2]. Column 5: Schweiger et al.[4]. Column 6: Our GRA. The original image in the last row is from the CIE report[22], while rest of the input images are from Kodak dataset[21].

gamut colors to far-away points on the destination gamut. Moreover, the HPMINDE algorithm often yield images with loss of texture, as it can be observed in rows 1 and 4 of Figure 8. The methods of LCLIP[3] (second column) and Schweiger et al.[4] (fifth column) may produce results with excessive desaturation in bright regions, as shown on the close-ups of the helmet and on the neck of the yellow parrot. POSTCLIP[18] may cause noticeable hue shifts in some images: see for example third column of both row 1 and row 4 of Figure 8. In the example of the second row of Figure 8, all tested GRAs except the proposed one produce tonal discontinuities on the face of the woman.

Zamir et al.[19] showed that the current image quality metrics, when applied to the gamut extension problem, provide results that do not correlate well with users' choices. Which is why, in this paper we are providing quantitative results only for the gamut reduction algorithms. For this, we use a
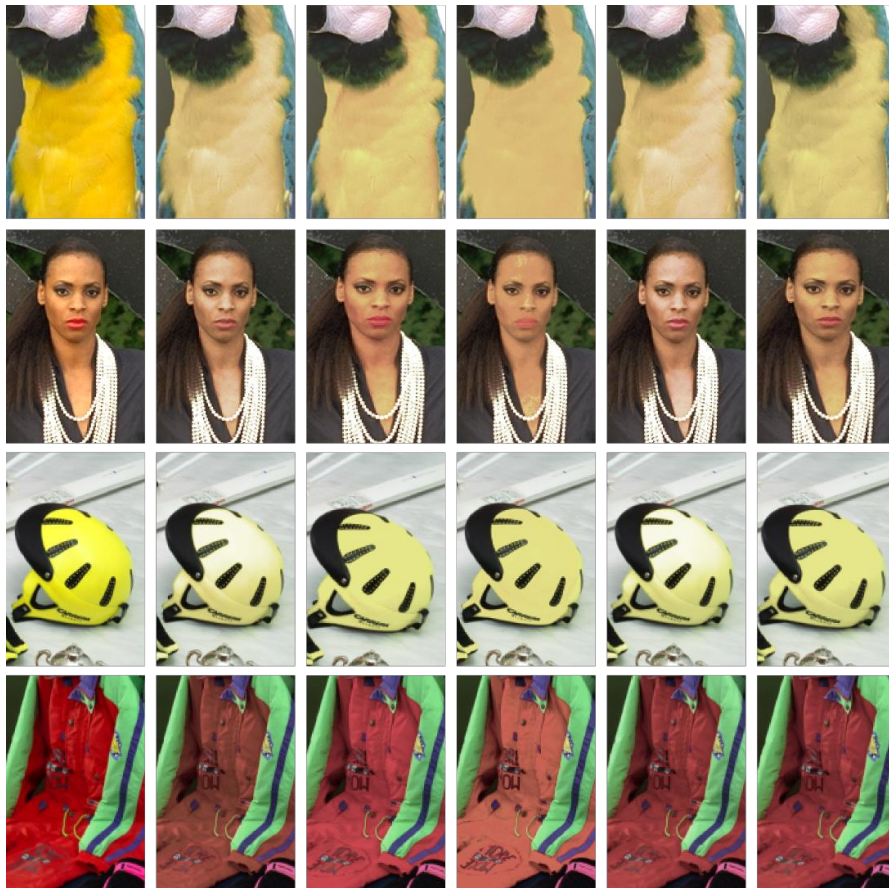

Figure 8. Comparison of GRAs: crops are from Figure 7. Column 1: input images. Column 2: LCLIP[3]. Column 3: POSTCLIP[18]. Column 4: HPMINDE[2]. Column 5: Schweiger et al.[4]. Column 6: Our GRA.

Table 2. Quantitative assessment using CID[20]: mean error across 31 images from[21-23].

	LCLIP[3]	POSTCLIP[18]	HPMINDE[2]	Schweiger et al.[4]	Our GRA
Mean Error	0.0225	0.0336	0.0491	0.0243	**0.0165**

Figure 9. One-shot versus iterative GRA: a trade-off between quality and speed. Left: input image. Middle: output of one-shot GRA obtained in 0.58 seconds. Right: output of iterative GRA obtained in 1.47 seconds. It can be seen that the colors reproduced by the iterative GRA are more saturated than those of one-shot GRA.

perceptual color image difference (CID) metric[20] which is particularly tailored to assess the quality of gamut reduction results. The CID metric compares the gamut-mapped image with the reference (original) image and analyzes differences in several image features such as hue, lightness, structure, chroma and contrast. We apply the competing GRAs on all 24 images of Kodak dataset[21] and on 7 other commonly used images in gamut mapping evaluation[22,23]. In Table 2 we summarize the mean errors where it can be seen that our GRA outperforms other methods.

As mentioned before, the method presented in Eq. (1) is capable of reducing the gamut of a given input image. However, this one-shot method will affect not only the out-of-gamut pixels but also those pixels that are already inside the target gamut, therefore yielding an output image with significantly reduced saturation. To deal with this issue, we make use of the proposed GRA in an iterative manner, which leaves in-gamut pixels unmodified and therefore providing more saturated results. However, this improvement in quality comes at the cost of more computational time. This trade-off between quality and speed is depicted in Figure 9.

Conclusion

We presented a gamut mapping framework that offers a functionality to perform both gamut reduction and gamut extension. The proposed framework is based on retinal and color perception models from vision science, and it produces results that are visually faithful to the original content. Moreover, the evaluation performed by using the perceptual CID metric showed that the results of our GRA are more similar to the original images than those of other competing methods. The

proposed gamut mapping algorithms are reliable and fast, and therefore have a great potential to be implemented in movie postproduction suites, in cameras, in display devices, and in simultaneous UHDTV and HDTV broadcasts.

We are currently working on automating the computation of the gamma parameter and optimizing the method in terms of speed, for instance by minimizing the number of iterations needed in the GR case. Moreover, we are exploring the effect of applying our methods on the saturation component computed in several different ways in the literature. We are also aiming to conduct psychophysical studies in order to evaluate the results for gamut reduction and gamut extension by using different displays (cinema projector, TV, and computer monitor) in different lighting conditions (cinema, home and office). All the examples in this paper have been obtained by running a non-optimized MATLAB implementation on a Desktop PC (12 cores 3.4-GHz CPU, 32GB RAM) so there is much room for improvement.

Acknowledgements

Appendix

One can transform RGB images into HSV color space by using the following set of equations[24]:

$$M = \max(R, G, B)$$

$$m = \min(R, G, B)$$

$$C = M - m$$

$$H' = \begin{cases} \text{undefined}, & \text{if } C = 0 \\ \dfrac{G - B}{C} \bmod 6, & \text{if } M = R \\ \dfrac{B - R}{C} + 2, & \text{if } M = G \\ \dfrac{R - G}{C} + 4, & \text{if } M = B \end{cases}$$

$$H = 60° \text{ x } H'$$

$$S = \frac{M - m}{M}$$

$$V = M$$

Refrences

[1]   J. Morovic, *Color Gamut Mapping*, John Wiley & Sons, 2008.

[2]   G. M. Murch and J. M. Taylor, "Color in Computer Graphics: Manipulating and Matching Color", Eurographics Seminar: Advances in Computer Graphics V, pages 41–47, 1989.

[3]   J. J. Sara, "The Automated Reproduction of Pictures with Nonreproducible Colors", PhD Thesis, Massachusetts Institute of Technology (MIT), 1984.

[4]   F. Schweiger, T. Borer, and M. Pindoria, "Luminance-Preserving Colour Conversion", SMPTE   Annual Technical Conference and Exhibition, pages 1–9, 2016.

[5]   R. Bala, R. Dequeiroz, R. Eschbach, and W. Wu, "Gamut Mapping to Preserve Spatial Luminance Variations", Journal of Imaging Science and Technology, 45:122–128, 2001.

[6]   S. W. Zamir, J. Vazquez-Corral, and M. Bertalmío, "Gamut Mapping in Cinematography Through Perceptually-Based Contrast Modification", IEEE Journal of Selected Topics in Signal Processing, 8:490–503, 2014.

[7]   J. A. Pytlarz, K.D. Thurston, D. Brooks, P. Boon, and R. Atkins, "Real Time Cross-Mapping of High Dynamic Range Images", IET Conference Proceedings, 2016.

[8]   C. Gatta and I. Farup, "Gamut Mapping in RGB Colour Spaces with the Iterative Ratios Diffusion Algorithm", In Proc. of IS&T/SPIE Electronic Imaging, pp 12–20, 2017.

[9]   S. E. Casella, R. L. Heckaman, and M. D. Fairchild, "Mapping Standard Image Content to Wide-Gamut Displays", In Proc. of Color and Imaging Conference, pp 106–111, 2008.

[10]   R. L. Heckaman and J. Sullivan, "Rendering Digital Cinema and Broadcast TV Content to Wide Gamut Display Media", SID Symposium Digest of Technical Papers, 42(1):225–228, 2011.

[11]   H. Pan and S. Daly, "A Gamut-Mapping Algorithm with Separate Skin and Non-Skin Color Preference Controls for Wide-Colorgamut TV", SID Symposium Digest of Technical Papers, 39:1363–1366, 2008.

[12]   J. Laird, R. Muijs, and J. Kuang, "Development and Evaluation of Gamut Extension Algorithms", Color Research & Application, 34:443–451, 2009.

[13]   S. W. Zamir, J. Vazquez-Corral, and M. Bertalmío, "Perceptually-Based Gamut Extension Algorithm for Emerging Wide Color Gamut Display and Projection Technologies", In SMPTE Annual Technical Conference & Exhibition, pages 1–11, 2016.

[14]   M. Bertalmío, V. Caselles, E. Provenzi, and A. Rizzi, "Perceptual Color Correction Through Variational Techniques", IEEE Transactions on Image Processing, 16(4):1058–1072, 2007.

[15]   J. Kim, T. Batard, and M. Bertalmío, "Retinal Processing Optimizes Contrast Coding", 10.7490/f1000research.1112400.1, 2016.

[16]   J. Froehlich, S. Grandinetti, B. Eberhardt, S. Walter, A. Schilling, and H. Brendel, "Creating Cinematic Wide Gamut HDR-Video for the Evaluation of Tone Mapping Operators and HDR-Displays", In Proc. of IS&T/SPIE Electronic Imaging, 2014.

[17]   S. Andriani, H. Brendel, T. Seybold, and J. Goldstone, "Beyond the Kodak Image Set: a New Reference Set of Color Image Sequences", In Proc. of IEEE International Conference on Image Processing, pp 2289–2293, 2013.

[18]   J. Froehlich, A. Schilling, and B. Eberhardt, "Gamut Mapping for Digital Cinema", In SMPTE Annual Technical Conference & Exhibition, pages 1–11, 2013.

[19]   S. W. Zamir, J. Vazquez-Corral, and M. Bertalmío, "Gamut Extension for Cinema", IEEE Transactions on Image Processing, 26(4):1595–1606, 2017.

[20]   I. Lissner, J. Preiss, P. Urban, M. S. Lichtenauer, and P. Zolliker, "Image-Difference Prediction: From Grayscale to Color", IEEE Transactions on Image Processing, 22(2):435–446, 2013.

[21]   Kodak. http://r0k.us/graphics/kodak/, 1993.

[22]   CIE, "Guidelines for the Evaluation of Gamut Mapping Algorithms", Technical Report, CIE 156, 2004.

[23]   ISO. 12640-2: Graphic Technology – Prepress Digital Data Exchange – Part 2: XYZ/sRGB Encoded Standard Colour Image Data (XYZ/SCID), 2004.

[24]   A. R. Smith, "Color Gamut Transform Pairs", In Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques, pages 12-19, 1978.