

# Detection and correction of unnatural purple colors in images of refractive objects taken by digital still camera

Mikhail Matrosov, Alexey Ignatenko

Department of Computational Mathematics and Cybernetics  
Lomonosov Moscow State University, Moscow, Russia  
{matrosov, ignatenko}@graphics.cs.msu.ru

Sergey Sivovolenko

OctoNus Software Ltd.  
sivovolenko@octonus.com

## Abstract

We discovered significant error in color in images produced by a digital still camera used to capture scenes with a special setup of illumination and an object. The object is small, transparent, color-less and has light-refractive properties. Illumination is produced by a number of LEDs with known spectra. Due to light dispersion in the object, vivid monochromatic colored flares can be seen and captured. Images from a digital still camera captured in the described setup occasionally exhibit bright purple (almost pink) colors, which do not correspond to any monochromatic color.

In this paper, we analyze the origins of this effect by examining different properties of the setup and analyzing RAW images. We propose a simple and efficient algorithm for correction of those unnatural purple colors by using only final JPEG image produced by the camera. In this algorithm, we develop a continuous transform which maps all unnatural colors to the natural ones in a perceptually uniform color space. Then, a 3D look-up table is constructed using this transform, which in turn is used for further correction of images captured by the camera.

**Keywords:** *color management, color calibration, color correction, monochromatic colors, RAW-processing, perceptually uniform color spaces, light dispersion, digital still camera.*

## 1 BACKGROUND

Consumer digital still cameras are very powerful tools for capturing real world images. They are broadly available, well-studied and intensively developed, that's why they are frequently used not only by photographers, but in many science and engineering applications. The latter require precise, repeatable and calibrated results.

Camera color calibration is not a trivial task since camera "sees" world in a model-specific color space. In order to be able to process and correctly display such an image one should be able to convert it to a certain conventional color space, like CIE XYZ [1]. The whole chain of color processing in digital cameras including this issue is briefly and clearly described by Adams et al. in [2]. They describe this conversion to be handled by a  $3 \times 3$  matrix converting camera-specific RGB response to universal XYZ values. The tricky part is that camera spectral sensitivities cannot usually be represented as linear combinations of CIE color matching functions forming XYZ values.

So this conversion matrix is usually designed to minimize averaged error for a specific set of colorants. Spaulding et al. [3] used Macbeth Color Checker [4] as a target set and an RMS error of CIELAB  $\Delta E_{ab}^*$  color-difference measure to find an optimal matrix:

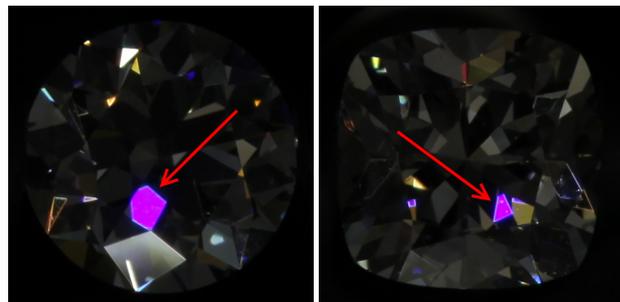
$$\Delta E_{RMS}^* = \sqrt{\sum_{i=0}^N (\Delta E_i^*)^2},$$

where  $N$  is the number of color patches and

$$\Delta E_i^* = \sqrt{(L_{si}^* - L_{di}^*)^2 + (a_{si}^* - a_{di}^*)^2 + (b_{si}^* - b_{di}^*)^2},$$

where  $L_{si}^*$ ,  $a_{si}^*$  and  $b_{si}^*$  are the CIELAB scene color values for the

Russia, Moscow, October 01–05, 2012



**Figure 1:** Images with regions of unnatural purple color for two different objects.

$i$ -th color path and  $L_{di}^*$ ,  $a_{di}^*$  and  $b_{di}^*$  are the CIELAB reproduced color values for the  $i$ -th color path.

Hong et al. [5] used a broader collection of colorants: an ANSI IT8.7/2 [6] chart on Kodak Ektacolor Professional Paper and the textile samples selected from The Professional Colour Communicator [7] using reactive dyes on cotton. They also performed a polynomial regression with least-squares fitting to minimize the color-reproduction error.

Thus, there are many available techniques to perform conversion of camera RGB response to XYZ values which can generally include some non-linear transformations or multidimensional look-up-tables. And we do not know precisely how a particular camera model handles this conversion since most of camera firmwares are proprietary and closed.

However, most of these techniques focus on reproduction of colors usually observable in natural scenes, but not all the physically available spectra. In specific engineering tasks, a certain spectra can be encountered, which a camera would not be able to handle properly. That is the case discussed in this paper.

In our task, we analyze optical properties of a transparent colorless object shaped as a polyhedron. Its refractive index is high enough to make it a dispersive media. Thereby, when illuminated by a white light, such an object appears to have colored faces. Since colors are induced by light dispersion and the object's faces are small enough, color spectra of a single face is nearly constant and virtually monochromatic. We used a consumer digital still camera to capture images of the described scene and discovered vivid saturated purple colors appearing under certain conditions. Examples of such images are shown in figure 1.

However, such vivid purple colors do not correspond to any monochromatic spectra and we weren't able to witness the same purple faces with naked eyes observation. Thus we have decided that we've encountered the mentioned above case of a camera being unable to properly represent captured color. We analyzed this issue and created simple a correction algorithm that works well with our setup. While the algorithm is very specific and aimed at our particular task, the conducted research is extensive and general enough.

## 2 INTRODUCTION

Detailed description of the used setup, including notes on illumination, objects' properties and camera model, is given in section 3. This section also contains information on how images were obtained and how a collection of analyzed images was formed.

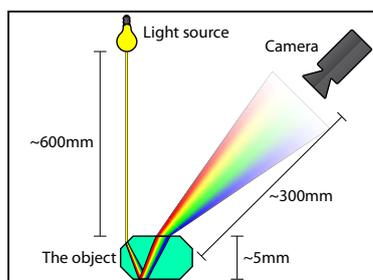
Extensive issue examination is done in section 4. We analyze the behavior of RGB response function in both output camera JPEG image and primary camera RAW image and show the evidence of error in color management process done by the camera in our specific case. We did not analyze the spectra of the faces which produced the unnatural purple colors since they are too small (about 100 microns in diameter) to be measured by any of the consumer spectrometers and appropriate measurement devices were unavailable for us. We do not consider unnatural colors to be induced by non-visible parts of spectra since we've done additional tests with UV and IR filters and the difference was barely noticeable.

Though we show that a consumer camera cannot properly handle discussed colors, we want to be able to use our setup for further research of described objects. So in section 5 we propose a simple correction algorithm which can be applied to JPEG camera output images to replace unnatural vivid purple colors with ones of a more bluish hue, which can be observed as a result of light dispersion. This algorithm continuously shifts specific hues in a perceptually uniform color space within boundaries estimated from captured collection of images. Examples of images corrected with the proposed algorithm are shown in section 6.

Conclusions and acknowledgements are given in sections 7 and 8 respectively.

## 3 SETUP AND PHOTOS

There are three essential components of the setup: illumination, an object and a camera. All of these are enclosed in a closed box with illumination mounted on the top, an object mounted at the bottom and a camera placed at the front and directed at the object. See figure 2 for schematic illustration<sup>1</sup>.



**Figure 2:** Schematic illustration of the setup used. Relative sizes of the objects and relative distances in the scene are not preserved for illustrative purposes.

Illumination consists of a number of bright LEDs with wide warm spectra. Camera white-balance was adjusted automatically prior to the shooting of any images. A sample paper patch with a neutral color was used for this purpose. There were about 50 LEDs, each of which is small enough and is supposed to approximate a point light source.

An object has a shape of a polyhedron with 50-70 faces and is 4-6 millimeters in diameter. It is made of a transparent colorless material with refractive index about 2.41, hence it introduces strong light dispersion and its faces appear to be colored when observed under appropriate illumination from a suitable point of view. The object is fixed on a motorized holder allowing rotation along two perpendicular axes situated in plane orthogonal to

<sup>1</sup>Camera icon designed by Go Squared Ltd.

camera's optical axis. Controlling this holder, one may adjust the position of the object in which a face with a color of interest will be observable by the camera.

The camera is mounted in front of the box and is pointed at the object. It is plugged into and is operated by a computer, so one can capture images of the object without touching the camera, which can lead to undesired vibrations of the box and break down the current dispersive pattern. In our tests we used a Canon EOS 5D Mark II digital still camera with a Canon EF 100mm f/2.8 Macro USM lens and a Kenko Teleplus PRO 300 DGX 1.4x AF teleconverter. However, as we will show later, explored effect poorly depends on a specific model of the digital camera.

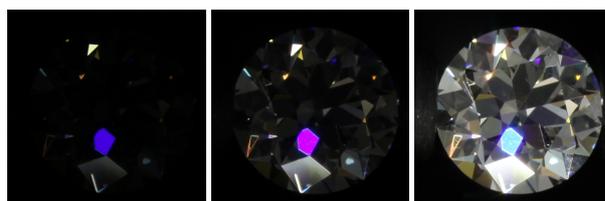
With the given setup, the linear size of an object on captured images came out to be 400-600 pixels. To obtain images of an object with purple faces (like the ones shown in figure 1) we simply rotated the holder slightly in an arbitrary direction and made a shot with the camera. We then studied acquired image for purple faces and optionally suggested a direction for further rotation. The effect is not quite rare, so usually it was enough to make 3-5 shots of the object to detect a new purple face and additional 2-3 shots to select an appropriate exposure.

Once a purple face was detected and an appropriate central exposure was selected, we made 11-15 shots of the same scene with different exposures using a  $\frac{2}{3}$  E.V. step. In other words, by making 15 shots we captured a number of images taken with exposures from  $-4\frac{2}{3}$  to  $+4\frac{2}{3}$  E.V. relative to the central exposure.

All in all, we captured 11-15 exposures for each of the 3-5 positions of 5 objects resulting in 254 images.

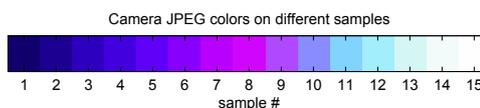
## 4 EXAMINATION

In this section we analyze separate sets with a number of images of the same scene taken with different exposures. Let us examine one particular set with 15 exposures; images in this set are numbered from 1 to 15 in order of the increasing exposure. Images #4, #8 and #12 are shown in figure 3. These are conventional output images of the camera stored as jpeg-files in the standard sRGB color space and we will refer to them as JPEG images (in contrast to RAW images, which will be discussed later).



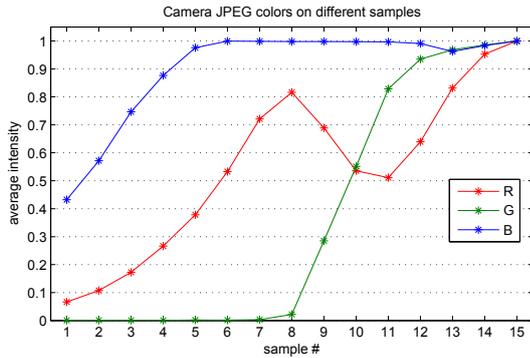
**Figure 3:** Three out of fifteen images in the examined set.

We begin analyzing the set by manually masking the region of the purple face on the images. Since all of the images are perfectly aligned, the same mask is used for all the images. We then average colors inside the mask for each of the images. In case there are several purple faces, we only mask the biggest one for simplicity and consistent averaging. Colors computed in such a way for all images in the set are shown in figure 4. We will refer to them as JPEG colors.



**Figure 4:** Averaged colors of a masked purple face for different images.

Since all these colors are from the same scene taken with different exposures, one can mention strange behavior of the color



**Figure 5:** Per-channel values of averaged colors of purple face for different JPEG images in the set.

transition with the increase of exposure. Blue color found in samples 1-4 changes to purple color in samples 5-9 and then to teal in samples 10-13, finally overexposing to white in samples 14-15. Only the last transition to white is expected, while others should not emerge in properly calibrated color-capturing device.

Intensity values of JPEG colors in the sRGB color space are plotted in figure 5. The first thing to mention is the non-monotonous R channel. Second is the different dynamic of growth of R and B channels in samples 1-5. Apparently, these facts indicate an unidentified error during capturing process.

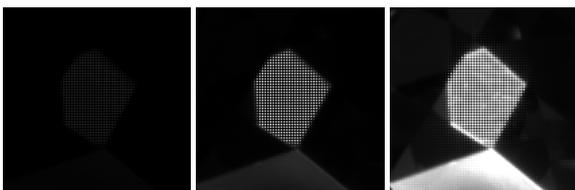
#### 4.1 Analyzing RAW

To dig into the origins of the problem, we analyze low-level responses of camera sensors stored in camera RAW images. Unfortunately, Canon EOS 5D Mark II provides RAW in CR2 format which is proprietary and not open. Moreover, none of the publicly available official Canon utilities like Digital Photo Professional or RAW Codec provide functionality to extract non-interpolated Bayer mosaic with primary sensor response. This is also true for a number of well-known RAW-processing utilities including Adobe Camera RAW, Adobe Lightroom, Apple Aperture and others.

Considering this fact, we used a free open-source utility dcrw (v9.12) with command line arguments “-d -4 -w” to obtain a non-interpolated Bayer mosaic from CR2 files. According to dcrw manual, the meaning of these arguments is as follows:

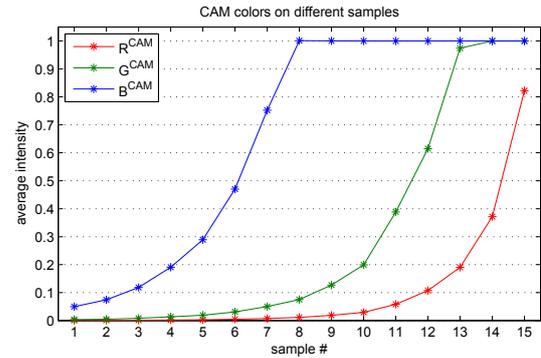
- d Document mode (no color, no interpolation)
- 4 Linear 16-bit
- w Use camera white balance, if possible

In this way we obtain linear responses from camera sensors in the form of pgm-files (see [8] for notes on PGM format) with 16-bit color depth and with white balance corrected using values estimated by the camera. Obtained colors are encoded in a camera-specific color-space, let us call it CAM. Likewise, we will refer to these images as CAM images. Cropped regions of a purple face from CAM images are shown in figure 6. They correspond to samples shown in figure 3.

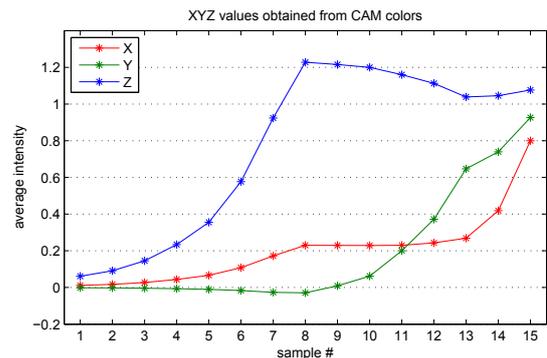


**Figure 6:** Linear Bayer mosaic extracted from RAW images.

Intensity values of CAM colors are plotted in figure 7. Averaging is done by splitting grayscale mosaic according to Bayer pattern



**Figure 7:** Per-channel values of averaged CAM colors.



**Figure 8:** XYZ values obtained with dcrw conversion matrix.

RGGB. The twin Bayer green components are treated equally. Analyzing this plot one can state the following observations:

- Values are linear within  $[0; 1]$  range.
- Values exceeding  $[0; 1]$  range are cropped.
- Values of channel  $B^{\text{CAM}}$  are much greater than values of other channels.
- Channel  $B^{\text{CAM}}$  saturates at image #8; channel  $G^{\text{CAM}}$  saturates at image #14.

Matching these observations with mentioned color transitions in JPEG images we can conclude the following:

- Color is changing from blue to purple until saturation of  $B^{\text{CAM}}$  channel.
- Once  $B^{\text{CAM}}$  channel is saturated, color is changing towards teal until saturation of  $G^{\text{CAM}}$  channel occurs.
- After that color is changing towards white.

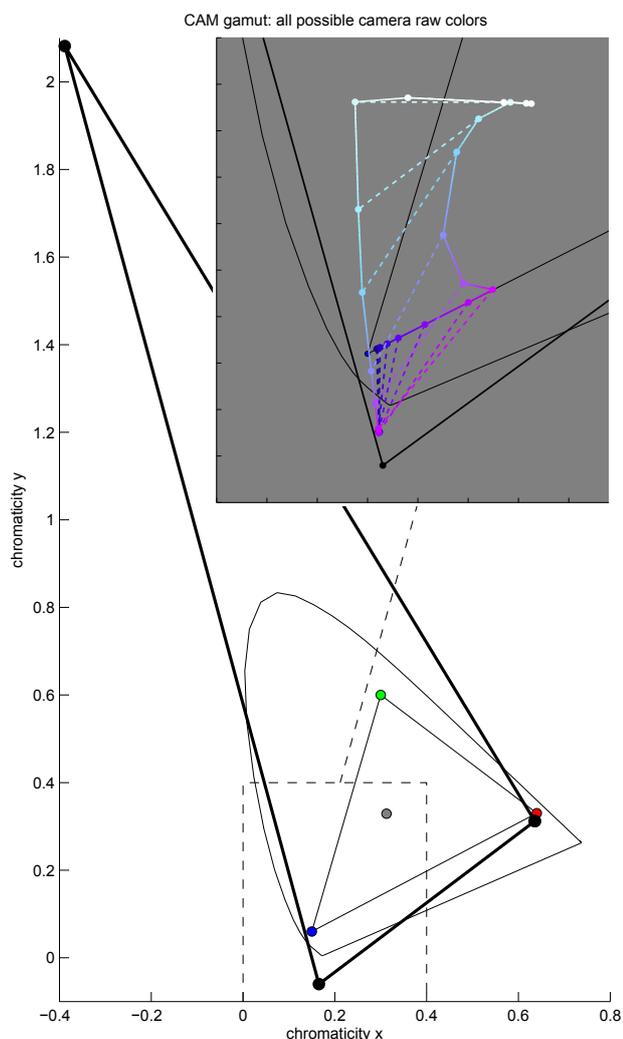
To be done with basic RAW analyzing, we make the conclusion that all unexpected color transitions correspond to the subsequent saturation of CAM color channels.

#### 4.2 Processing RAW

Since the camera converts a primary RAW image to a final JPEG image using proprietary closed firmware, we cannot precisely reproduce this process. However, we do know that it performs demosaicing to interpolate Bayer pattern and it uses a certain conversion matrix to obtain XYZ values from CAM values. Such a matrix is represented in dcrw source codes. Apparently, it was acquired by some reverse-engineering. We can use this matrix to convert CAM values to XYZ values, which are given in figure 8.

By analyzing this plot we conclude the following:

- XYZ values are linear while CAM values lay within  $[0; 1]$  range.
- When CAM values exceed  $[0; 1]$  range, XYZ values begin to change nearly unpredictably.



**Figure 9:** Gamut of CAM colors depicted on xyY chromaticity diagram against sRGB triangle. Zoomed part shows pairs of JPEG colors (inside sRGB gamut) and corresponding CAM colors (outside sRGB gamut).

- Y value is negative while CAM values lay within [0; 1] range.

The latter point is crucial. Y value of an XYZ triplet cannot be negative by definition. Thus we consider CAM→XYZ conversion unreliable. The conversion is optimized to minimize errors in reproduction of normally observable colors and fails to properly handle values induced by purple faces. Applying this conversion to all possible CAM colors we derive its gamut, which is shown in figure 9. Zoomed part of the figure shows pairs of JPEG colors against CAM colors (background is grayed out to make luminous colors visible). Dashed lines connect corresponding elements of a pair. Solid lines connect colors from adjacent images for both CAM and JPEG series, thus creating two tracks.

It may be seen that camera’s rendering intent tends to make sRGB colors from given XYZ values more purple while it should be probably mapped to blue. This intent is the final reason why we see those unnatural colors on JPEG images. However, rendering intent is not of a big concern since XYZ values themselves are obtained using an unreliable CAM→XYZ conversion.

## 5 CORRECTION

By speaking of “correction of purple colors” we mean a scheme which allows to obtain conventional image of the scene without mentioned unnatural colors. Generally, it can include some kind

of RAW-processing, installation of color filters on a camera or certain manual adjustments. But we want this scheme to satisfy two requirements.

Firstly, a JPEG image produced by a camera has satisfactory quality, so we demand the corrected image to be as close to JPEG image as possible (except for regions of purple faces). Secondly, we want this scheme to be fast, meaning that the correction should only take a small fraction of time required to perform a shot.

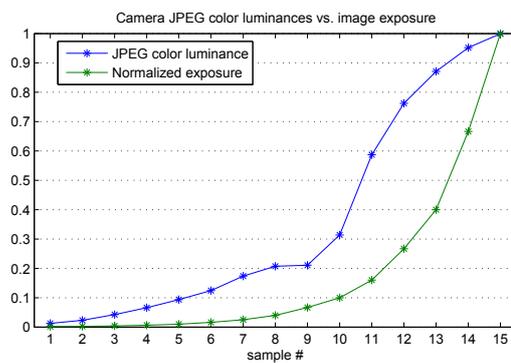
RAW-processing done by camera consists of a number of proprietary hardware-specific algorithms such as demosaicing, sharpening and postprocessing. It is unlikely that we can reproduce all of them in a simple and efficient enough RAW-processing line. So, in our scheme we propose to correct the resulting camera JPEG image itself.

Since we don’t reliably know the XYZ values of the purple faces’ colors, we can chose the corrected color somewhat arbitrarily. We suggest to simply shift the color hue of purple faces towards blue. The exact boundaries and directions of the shift are discussed further.

### 5.1 Proposed algorithm

Since in our task the only meaningful information in the camera JPEG image is the picture of an object, any purple color in any region of the image can be considered unnatural. Thus, we can apply a global correction based only on values of the particular pixel. Moreover, we can generate a 3D look-up-table to map values of input pixels to values of corrected pixels. This LUT transformation is very fast and since LUT itself can be pre-computed, we don’t have to worry about the performance of the transformation algorithm.

Now let’s deal with the properties of the color shift. Figure 10 illustrates luminance of JPEG colors plotted against normalized exposure for every image in the set. It may be seen that luminance is not linearly dependent on exposure. However, luminance plot is monotonous and its dynamics is similar to the dynamics of the exposure plot. So we will preserve the luminance value during the correction shift.



**Figure 10:** Luminance of JPEG colors against exposure.

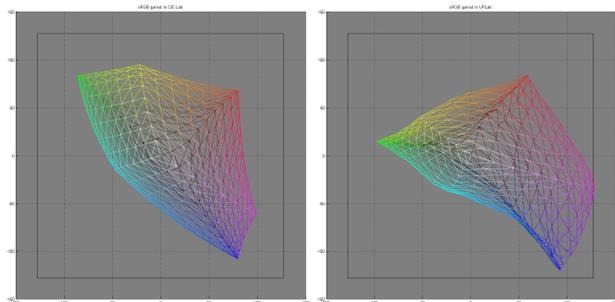
Since it is impossible (within the sRGB color space) to preserve both luminance and saturation while varying hue, we will have to compress the color saturation during the shift. Compressing saturation of the blue color is related to a known issue called the “blue turns purple” problem [9, The “Blue Turns Purple” Problem<sup>2</sup>]. We don’t want to shift purple towards blue only for it to become purple again after the compression of the saturation. In order to avoid this issue, we use the Perceptually Uniform Lab color space [9, Uniform Perceptual Lab<sup>3</sup>] for all of our color manipulations. Moreover, the exact rules controlling the hue shift

<sup>2</sup><http://brucelindbloom.com/MunsellCalcHelp.html#BluePurple>

<sup>3</sup><http://brucelindbloom.com/UPLab.html>

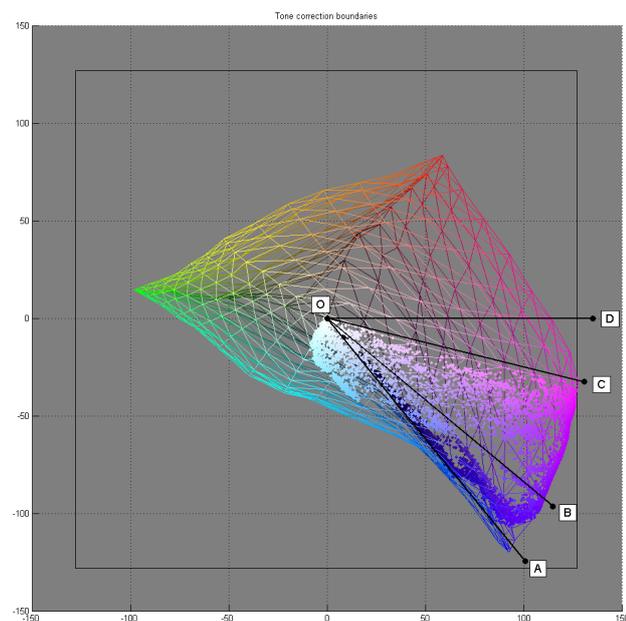
are much simpler in the UPLab color space than in the CIE Lab color space.

Rotation of the color radius-vector in UPLab within the  $ab$  plane corresponds to the Munsell hue [10] shift. Alteration of the radius-vector length corresponds to the shift in the Munsell chroma. We move pixels within the  $ab$  plane in the UPLab space preserving the  $L$  coordinate to keep luminance unchanged. Gamuts of the sRGB color space in CIE Lab and UPLab are shown in figure 11.



**Figure 11:** sRGB gamut in CIE Lab (left) and UPLab (right) color spaces (projection to  $ab$  plane with  $L$  axis directed up perpendicular to the image).

We divide the  $ab$  plane with four rays outgoing from the origin to four sectors of hues. These rays are shown in figure 12 as  $OA$ ,  $OB$ ,  $OC$  and  $OD$ . Colors of the purple faces collected through the entire collected base of images are also shown in this figure.



**Figure 12:** Color correction boundaries in UPLab. The points depict all the colors of purple faces from all the images from the entire base.

Let the color hue be in the range  $XY$  if its  $ab$  coordinates lie inside a sector formed by the  $OX$  ray moving to the  $OY$  ray counter-clockwise. Then hues of all possible colors are in one of the ranges  $AB$ ,  $BC$ ,  $CD$  or  $DA$ .

For a given pixel, correction shift depends on its hue:

- Hues in the  $DA$  range are not changed.
- Hues in the  $AC$  range are shrank to the  $AB$  range.
- Hues in the  $CD$  range are expanded to the  $BD$  range.

By shrinking/expanding one range to another we mean the following. Let  $P$  be a point in the  $ab$  plane in the UPLab space

Russia, Moscow, October 01–05, 2012

corresponding to the given color (with a persistent  $L$  coordinate). Let the prolongation of the  $OP$  ray intersect the sRGB gamut at point  $R$ . Let the correction move  $P$  to  $P'$  with prolongation of  $OP'$  intersecting the sRGB gamut at point  $R'$ . Then, if the color hue in the  $XY$  range is shrank/expanded to the  $X'Y'$  range:

$$\frac{\angle XOP}{\angle XOY} = \frac{\angle X'OP'}{\angle X'OY'} \quad \text{and} \quad \frac{|OR|}{|OP|} = \frac{|OR'|}{|OP'|}.$$

The  $A$  boundary passes near a distinctive cluster of blue colors on the sRGB gamut. The  $B$  boundary specifies the strength of correction and passes near the “most purple” observable monochromatic color. It may be varied to achieve the most pleasing result. The  $C$  boundary is selected so that all colors of purple faces collected through the entire image base lie within the  $AC$  range. The  $D$  boundary is selected somewhat arbitrarily to space the ranges evenly and to lie within the purple line of colors so that its hue would not represent any monochromatic color.

The proposed correction algorithm includes intersection of rays with 3D gamuts and conversion from the sRGB color space to the UPLab color space and backwards. These tasks require significant time to be performed, but since we precompute the LUT, they do not affect overall efficiency of the correction algorithm. It took us about 5 minutes to construct a LUT for all possible 16,777,216 sRGB colors. We used CGAL AABB Trees [11, 3D Fast Intersection and Distance Computation (AABB Tree)<sup>4</sup>] to compute intersections of rays with gamuts. The size of full LUT is 48MiB, but it can be easily compressed to  $\approx 7$ MiB using a run-length-encoding technique since most of the colors are unaffected by the correction and it can still be efficiently accessed in the compressed form.

## 6 RESULTS

To test the proposed correction algorithm, we prepared three correction LUTs differing in strength, i.e. with different locations of the  $B$  boundary. The feasibility of the algorithm was estimated manually by analyzing results for different samples from the base. Results for two objects under two different exposures are shown in figure 13 with close-up on purple faces. Results for other images from the base are available in the internet<sup>5</sup>.

Note how the rest of the image remain unchanged by correction and how purple faces preserve small details, which are of particular interest since they can be used to estimate clarity of the object. It is also visible, that the color of the faces in the images taken with lower exposure remains unaffected by the correction since it has the natural blue hue.

## 7 CONCLUSION

During our engineering work we have stumbled upon limitations of applicability of digital still cameras, where specific colors present in our scene could not be reproduced properly. We have analyzed sources of this issue and proposed a very efficient algorithm for its correction.

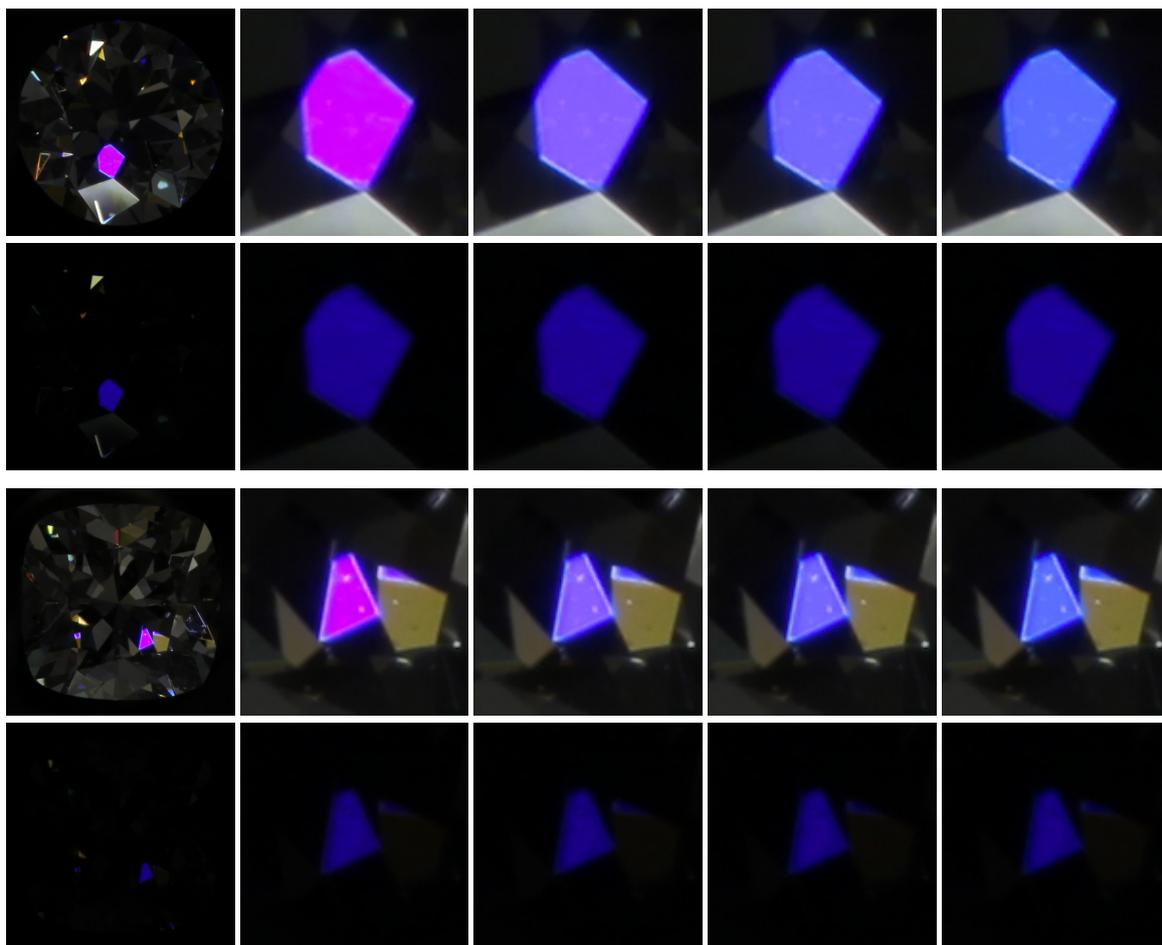
The study of the issue was conducted without access to the spectra of the color of interest because of a small physical size of an examined object. While proposed algorithm can be improved for more accurate color correction and developed to handle more general scenes, it suits our needs well and allows us to address the camera’s limitations.

## 8 ACKNOWLEDGEMENTS

This work was done in cooperation with and with financial and technical support of OctoNus Software Ltd.

<sup>4</sup>[http://www.cgal.org/Manual/latest/doc.html/cgal\\_manual/AABB\\_tree/Chapter\\_main.html](http://www.cgal.org/Manual/latest/doc.html/cgal_manual/AABB_tree/Chapter_main.html)

<sup>5</sup><ftp://graphics.cs.msu.ru/projects/PurpleFires/2012-04-21-MSS/all.html>



**Figure 13:** Examples of color correction done by proposed algorithm for two different objects, two rows per object. Odd rows contain images of the higher exposure (#8 in set) with the clearly visible purple face, even rows contain corresponding images with the lower exposure (#2 in set) with the same face of the natural blue color. The leftmost image in each row is the initial JPEG image taken by the camera, the second image to the right is a close-up of the purple face in this image. The remaining three images are close-ups of the same face in images corrected with different strength settings, i.e. with different locations of the  $B$  boundary. Correction strength increases from left to right.

## 9 REFERENCES

- [1] T. Smith and J. Guild, "The CIE colorimetric standards and their use," *Transactions of the Optical Society*, vol. 33, pp. 73–1931.
- [2] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *Micro, IEEE*, vol. 18, no. 6, pp. 20–30, 1998.
- [3] K.E. Spaulding, R.M. Vogel, and J.R. Szczepanski, "Method and apparatus for color-correcting multi-channel signals of a digital camera," Sept. 8 1998, US Patent 5,805,213.
- [4] C.S. McCamy, H. Marcus, and JG Davidson, "A color-rendition chart," *J. App. Photog. Eng*, vol. 2, no. 3, pp. 95–99, 1976.
- [5] G. Hong, M.R. Luo, and P.A. Rhodes, "A study of digital camera colorimetric characterisation based on polynomial modelling," 2001.
- [6] IT ANSI, "7/2-1993 (ISO 12641)," *Graphic Technology-Color Reflection Target for Input Scanner Calibration*.
- [7] J. Park and K. Park, "Professional colour communicator-the definitive colour selector," *Journal of the Society of Dyers and Colourists*, vol. 111, no. 3, pp. 56–57, 1995.
- [8] "PGM format specification," <http://netpbm.sourceforge.net/doc/pgm.html>, Accessed: 27/05/2012.
- [9] "Bruce lindbloom's web site," <http://brucelindbloom.com>, Accessed: 27/05/2012.
- [10] S.M. Newhall, D. Nickerson, and D.B. Judd, "Final report of the OSA subcommittee on the spacing of the munsell colors," *JOSA*, vol. 33, no. 7, pp. 385–411, 1943.
- [11] "CGAL - computational geometry algorithms library," <http://www.cgal.org>, Accessed: 27/05/2012.

## ABOUT THE AUTHORS

Mikhail Matrosov is a Ph.D. student at Moscow State University, Department of Computational Mathematics and Cybernetics. His research interests include interactive image enhancement, 3D reconstruction, HDR tone mapping, color theory and adjacent fields. His contact email is [matrosov@graphics.cs.msu.ru](mailto:matrosov@graphics.cs.msu.ru).

Alexey Ignatenko is a researcher at Computational Mathematics and Cybernetics department of Moscow State University. His research interests include photorealistic 3D rendering, 3D modelling and reconstruction, image-based rendering and adjacent fields. His contact e-mail is [ignatenko@graphics.cs.msu.ru](mailto:ignatenko@graphics.cs.msu.ru).

Sergey Sivovolenko is the CEO of OctoNus Software Ltd. His contact e-mail is [sivovolenko@octonus.com](mailto:sivovolenko@octonus.com).