Locally Adapted Detection and Correction of Unnatural Purple Colors in Images of Refractive Objects Taken by Digital Still Camera

Mikhail Matrosov[1], Alexey Ignatenko[1], and Sergey Sivovolenko[2]

[1] Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University, Moscow, Russia
{matrosov,ignatenko}@graphics.cs.msu.ru
[2] 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. Setup includes several LEDs as point light sources and a light-refractive object. Due to light dispersion in the object, vivid monochromatic colored flares appear. However, images captured with a digital still camera 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 unnatural purple colors by using only a final JPEG image produced by camera. We develop a continuous transform which maps all unnatural colors to the natural ones in a perceptually uniform color space. We also propose a simple segmentation technique to identify image areas to be corrected.

Keywords: color management, color calibration, segmentation, 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. Since they are broadly available, well-studied and intensively developed, the cameras are frequently used not only by photographers, but also in many research and engineering applications. The latter applications require precise, repeatable and calibrated results.

Color calibration of a camera is not a trivial task, since camera perceives world in a model-specific color space. In order to be able to process and correctly display such an image, it is necessary to convert it to a certain conventional color space, such as CIE XYZ [12]. The complete chain of color processing in digital cameras including this component is briefly and clearly described by Adams et al. [5]. They describe this conversion to be handled by a 3×3 matrix converting

M.L. Gavrilova et al. (Eds.): Trans. on Comput. Sci. XIX, LNCS 7870, pp. 117–130, 2013.

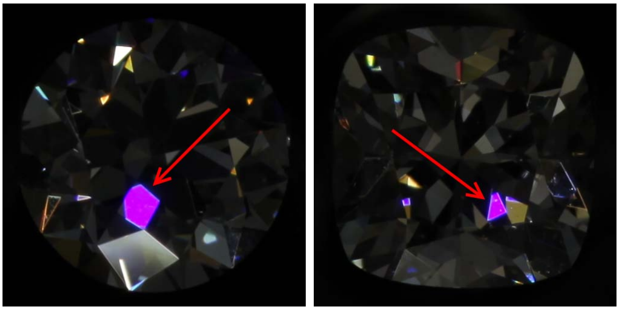

Fig. 1. Images with unnatural purple color for two different objects

camera-specific RGB response to universal XYZ values. The tricky part is that camera spectral sensitivities cannot be represented as linear combinations of CIE color matching functions forming XYZ values.

Therefore, this conversion matrix is usually designed to minimize the average error for a specific set of colorants. Spaulding et al. [13] used Macbeth Color Checker [9] as a target set and an RMS error of CIELAB ΔE^*_{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 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. [7] 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 [11] 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 are focused on reproduction of colors usually observable in natural scenes, but not of 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.

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 with 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 above mentioned case of a camera being unable to properly represent captured color. We present a simple correction algorithm to work with our setup. While the algorithm is very specific and aimed at our particular task, the conducted research is extensive and general.

2 Introduction

The detailed description of our setup, including the notes on illumination, properties of object 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.

An examination of the issue was done in our previous work [8] and is briefly recalled in section 4.

Though we prove that a consumer camera cannot properly handle discussed colors, we want to use our setup for further research of the 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. Firstly, we briefly recall our previous algorithm, and secondly, we propose a number of enhancements to it. The main enhancement is the segmentation technique, used to determine areas in which the correction is applied, as opposed to globally applied previous approach.

Examples of images corrected with the proposed algorithm are shown in section 6. Conclusions and acknowledgments are given in section 7.

3 Setup and Photos

There are three essential components of our setup: illumination, an object and a camera. All of these components 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 to the object. Figure 2 represents the schematic illustration[1].

[1] Camera icon designed by Go Squared Ltd.

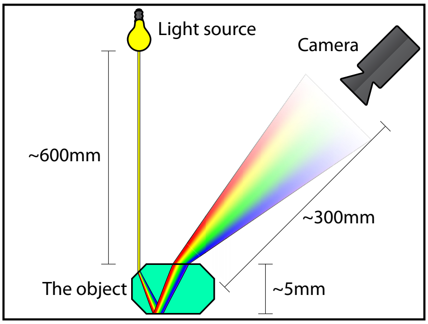

Fig. 2. Schematic illustration of the used setup. Relative sizes of objects and relative distances in the scene are not preserved for illustrative purposes.

Illumination consists of a several bright LEDs with wide warm spectra. The 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, which allows rotation along the two perpendicular axes, situated in the plane orthogonal to optical axis of a camera. 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, the 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 becomes 400-600 pixels. To obtain images of an object with purple faces (like the ones shown in figure 1) we rotated the holder slightly in an arbitrary direction and made a shot with the camera. We then studied the acquired image for purple faces and optionally suggested a direction for further rotation. The effect is not rare, therefore it is sufficient 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.

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

4 Examination

A thorough examination of the effect is done in our previous work [8]. This section briefly summarizes it.

One particular set with 15 exposures is examined, images in this set are numbered from 1 to 15 in order of the increasing exposure. The purple faces are manually marked up for this set and the colors within a single image are averaged, as shown in figure 3.

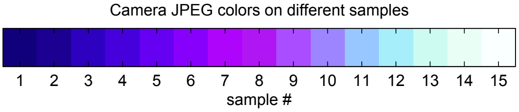

Fig. 3. Averaged colors of a masked purple face for the set of images with the increasing exposure

We then analyze low-level responses of camera sensors stored in camera RAW images. We use the free open-source utility dcraw [3] to obtain a non-interpolated Bayer mosaic in an unknown camera-specific CAM color space. We then conclude that the RAW images are linear on the exposure and the purple faces effect arises at a latter point.

Thus, we consider that the CAM→XYZ conversion is unreliable. As mentioned in section 1, the conversion is optimized to minimize errors in reproduction of normally observable colors and fails to properly handle values induced by purple faces.

5 Correction

In our previous work [8], we proposed a simple and efficient algorithm for correction of purple faces. However, due to its simplicity, it is unable to correctly handle a number of scenarios encountered in the real production environment. Several examples are given in figure 4, accompanied with results of the new proposed algorithm.

In this section we briefly recall the previous algorithm. We then propose several enhancements to it. The most significant enhancement is the introduced segmentation technique, which makes the algorithm locally adapted. Instead of applying the correction to the whole image, we correct only the areas marked as potential purple faces.

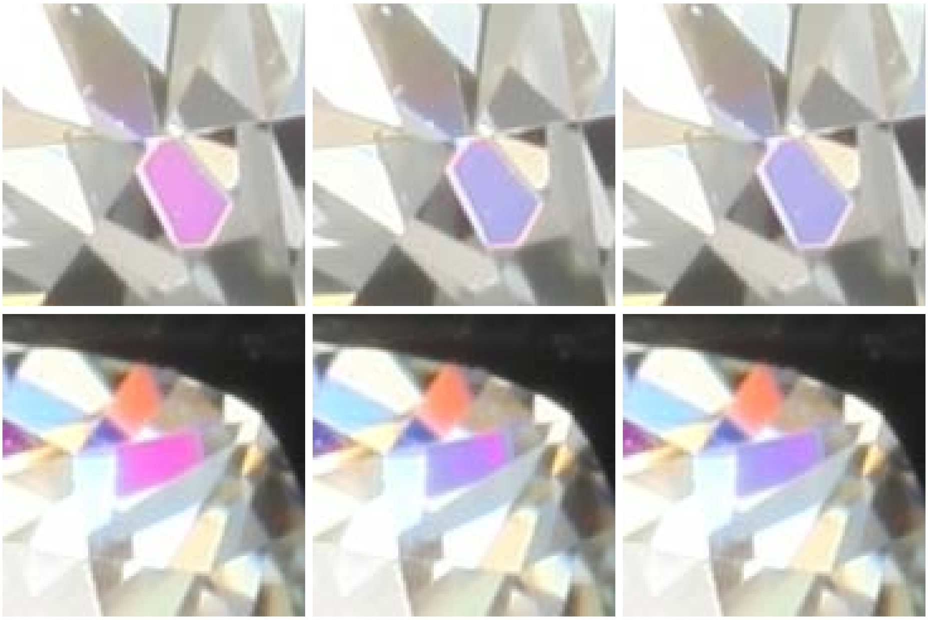

Fig. 4. The comparison between the proposed method (in the right column) and the previous method [8] (in the middle column). Two cropped source images are in the left column. Note the visual artifacts at the borders of the purple faces, produced by the previous method.

5.1 Previous Algorithm

First, we decide to work only with the final JPEG image produced by the camera, hence the correction is applied to this image, and no processing of the RAW file is done.

Then we propose to shift hue of a color and preserve its luminance. For this purpose we are using the Perceptually Uniform Lab color space [1, Uniform Perceptual Lab[2]], or simply the UPLab.

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.

We divide the *ab* plane with four rays outgoing from the origin to four sectors of hues. These rays are shown in figure 5 as OA, OB, OC and OD.

Let the color hue be in the range XY if its *ab* coordinates lie inside a sector formed by the OX ray, which moves 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:

[2] http://brucelindbloom.com/UPLab.html

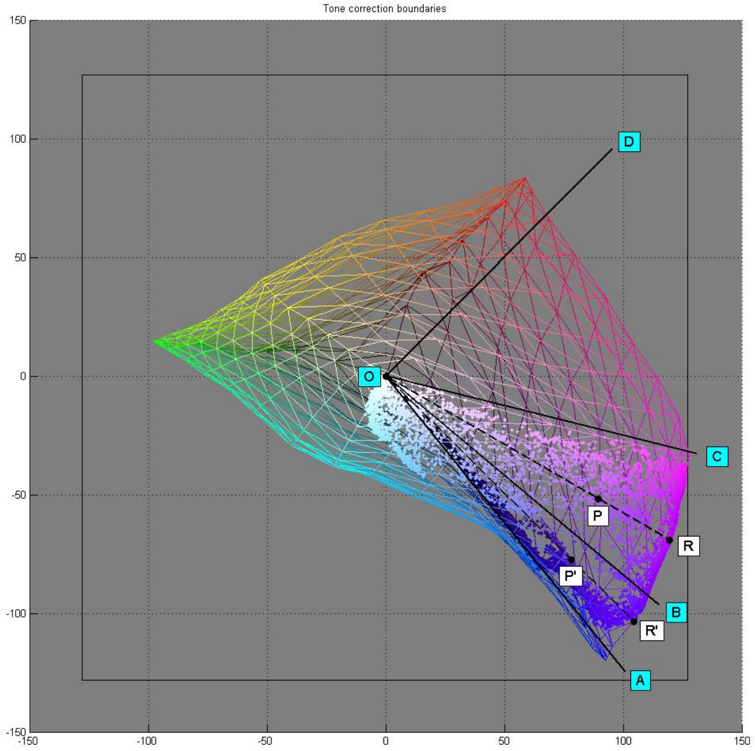

Fig. 5. The sRGB gamut in the UPLab with a several objects. The points depict all of the colors from purple faces in all images from the entire base. The teal marks correspond to correction boundaries. The white marks illustrate the correction.

- 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 stretched to the BD range.

By shrinking/stretching one range to another we imply the following. Let P be a point in the ab plane in the UPLab space corresponding to the given color (with a persistent L coordinate). Let the prolongation of the OP ray intersect the sRGB gamut at the point R. Let the correction move P to P' with prolongation of OP' intersecting the sRGB gamut at the point R' (see figure 5). Then, if the color hue in the XY range is shrank/stretched to the $X'Y'$ range, the following equations are met:

$$\frac{\angle XOP}{\angle XOY} = \frac{\angle X'OP'}{\angle X'OY'} \quad \text{and} \quad \frac{|OR|}{|OP|} = \frac{|OR'|}{|OP'|}, \tag{1}$$

which gives us

$$|OP'| = |OP|\frac{|OR'|}{|OR|}. \tag{2}$$

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. The C boundary is selected in such a way that all colors of purple faces collected through the entire image base lie within the AC range. The D boundary is selected somewhat arbitrarily and has different locations for the previous and the present works.

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 [2, 3D Fast Intersection and Distance Computation (AABB Tree) [3]] to compute intersections of rays with gamuts. The size of full LUT is 48MiB, but we compressed it to ≈7MiB 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.

5.2 Proposed Algorithm

In the present work, we propose a number of enhancements to our previous work. The main enhancement is to use a segmentation technique to select areas in which the correction needs to be applied. It is different from our previous approach, in which we applied the correction globally to the whole image.

The other two enhancements are: expanded CD range and constraint on the OP' length, i.e. the chroma of the corrected color. We discuss these two enhancements in current section, and leave the segmentation technique for a separate section.

We also optimized the exact location of the B boundary to be the average of the three possible locations presented in the previous work.

Expanded Range. By analyzing collected data, we conclude that the previous algorithm tends to perturb smooth gradients in a number of situations. The simplest way to make the correction smoother is to expand the stretched range of colors, i.e. the CD range.

As mentioned in our previous work, the exact location of the D boundary was chosen somewhat arbitrarily. We tried to make the whole range of affected colors considerably smaller to minimize potential distortions of the image. In the current work we place the D boundary at 45° instead of 0° (see figure 5).

After this modification the correction affects more colors, especially close to reds. However, affected colors still lie along the purple line and should not interfere with monochromatic colors.

[3] `http://www.cgal.org/Manual/latest/doc_html/cgal_manual/AABB_tree/Chapter_main.html`

In practice, there are colors of almost any hue on the production images. Hence it is important to preserve colors that we are sure should not be corrected. This is achieved by a segmentation technique discussed below.

Constrained Chroma. The sRGB gamut in the UPLab color space has a complex shape. This is especially pronounced near the magenta-white edge of the RGB cube. It is where the colors of the purple faces are located. After applying the equations (1), $|OP'|$ may be significantly larger than $|OP|$. That means, that the chroma of the color is increased.

We discovered that this effect contributes to the problem of perturbed gradients. Hence we constrain the chroma not to be enlarged by the correction. We replace the equation (2) with the following:

$$|OP'| = |OP| \min\left(1, \frac{|OR'|}{|OR|}\right).$$

5.3 Segmentation Technique

The above correction is applied only to areas of the image marked by a segmentation as possible regions with purple faces. The segmentation produces a binary decision mask, which marks every pixel on the image as to be either corrected, or ignored.

All the colors are divided into three classes:

- Strong purples
- Weak purples
- Others

The strong purples are the colors for which we are sure that they correspond to a purple face. To form these, we compute distance to the colors from marked up purple faces throughout the entire image base. Note that every purple face becomes black on very low exposures and white on very bright exposures. However, black and white are neutral colors and can be encountered anywhere on the image. Therefore we include additional constraint on chroma when we select strong purples.

The weak purples are the colors from the whole range affected by the correction, i.e. the colors from the *AD* range, except the colors that are marked as strong purples. They need to be processed to smooth transitions between strong purples and the rest of the image.

Finally, *the others* are the colors that are not affected by the correction under any circumstances. These are the colors lying outside the correction boundaries, the *AD* range.

After colors of all the pixels on the image are classified to one of these three classes, a segmentation mask is constructed. It includes all the connected regions of weak purples, for which there is a neighboring strong purple. That is, strong purples act as a seed pixels for a flood-fill, constrained with weak purples, as follows.

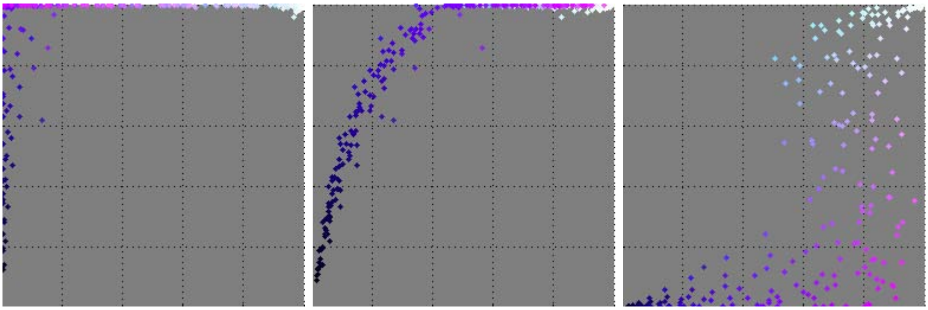

Fig. 6. Projections of median colors of the purple faces on the GB, RB and RG planes of the RGB cube, from left to right. The coordinates are in the sRGB color space.

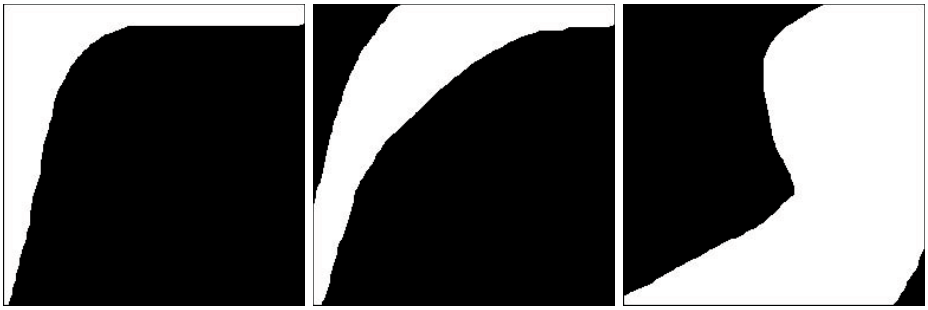

Fig. 7. The masks used for detection of strong purples. A color is considered close to the colors of purple faces if and only if its projections on the GB, RB and RG planes of the RGB cube correspond to white pixels of the masks, from left to right. The outer black strokes are added for visualization only.

Detection of Strong Purples. For every marked up purple face from the entire image base, median color is computed. All these median colors are placed in the RGB cube for the source sRGB color space. They are then projected on the different sides of the RGB cube. The projections are shown in figure 6.

Our task is to define a rule which can determine whether a given color lies within a cluster of colors formed by purple faces. This rule doesn't have to be very precise, since the selected strong purples only act like seeds.

We propose a simple solution for this rule. We manually compose masks for the three projection planes. We then assume, that a color lies within a cluster of colors formed by purple faces if and only if its projections on the planes of the RGB cube lie within the composed masks. These masks are shown in figure 7.

Finally, a color is considered a strong purple, if and only if:

- Its projections on the sides of the RGB cube lie within the given masks
- It lies within the AC range
- Its chroma is greater than 50

The first condition is computed in the sRGB color space. The remaining two are computed in the UPLab color space. For the sake of performance, a technique utilizing a look-up table is used, similar to the one used for the correction. All the sRGB colors are checked to be strong or weak purples, and the results are stored in 3D LUTs. The details about this process are given below.

Construction of the Mask. After colors of all the pixels on the image are classified into strong purples, weak purples, or others, the final segmentation mask is constructed.

Pixels with colors classified as weak purples are divided into connected components. A component is included in the mask if and only if it has a neighboring pixel with a color classified as strong purple. The process is illustrated in figure 8.

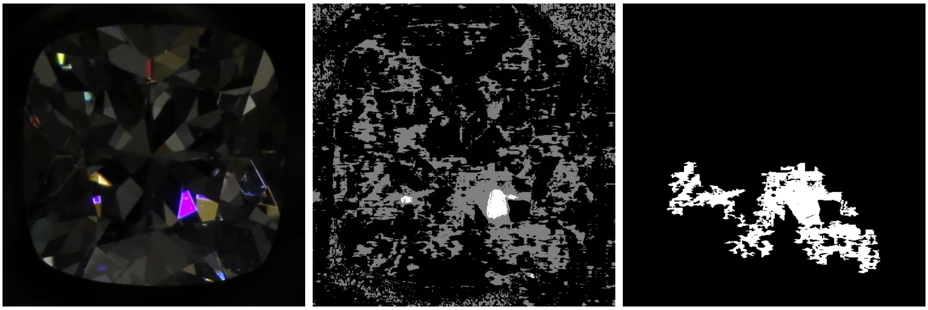

Fig. 8. Construction of a segmentation mask. The source image is shown in the left. Classification result is shown in the middle, with strong purples marked as white, weak purples marked as gray, and others marked as black. The constructed segmentation mask is shown in the right.

As one can see, the constructed mask is not very accurate or discriminative. But this is not important, since most pixels accidentally included in the mask are near-neutral. We only want to preserve chromatic colors while correcting purple faces. Neutrals are very slightly affected by the correction.

In order to reduce the noise impact, the mask of strong purples is morphologically eroded before the segmentation mask is constructed. A disk with a 2-pixel radius is used as a structural element for the erosion. The pixels excluded from the mask of strong purples in such a manner, are marked as weak purples.

Implementation. To accelerate the classification process we precompute the results for all sRGB colors. Two look-up tables are constructed at this stage: one for strong purples, and one for weak purples. It is important to mention that the table with weak purples also contains strong purples. It makes the implementation easier.

These two tables are compressed using the same RLE-technique as the correction table. Additionally, since the masks are binary, they are stored as bit-encoded, with every byte describing eight neighboring colors. Using this encoding we reduced the total size of the tables from 32MiB to 1MiB (250KiB and 850KiB for strong and weak purples respectively).

The connected components are extracted using a simple flood-fill algorithm, started from each pixel marked as a strong purple. The flood-fill is spreading only on pixels marked as weak purples.

The morphological erosion and flood-fill are done using the corresponding functions from the highly optimized Intel® IPP library [4].

6 Results

To ensure that the proposed enhancements contribute to the quality of the results, we manually selected 33 samples from production photos, in which the previous algorithm exhibited visual artifacts. We then manually checked the results on all of the samples. Three of the samples are shown in figure 4. The others can be found in the supplementary materials available on the Internet[4].

Several examples of correction made by the proposed algorithm are given in figure 9. They are not very different from the ones generated by the previous algorithms. For differences see figure 4.

The introduction of local processing slowed the algorithm by 7 times. However, it is still very fast and it takes only 430ms to process a 21Mpix photo on Intel® E8500 CPU running two cores at 3.16GHz.

7 Conclusion

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

In the present work we proposed a several enhancements to the previous algorithm. With these enhancements, the quality of the algorithm is sufficient for processing of production photos. Despite the proposed enhancements complicated the algorithm, it is still very efficient.

The further enhancement of the algorithm may take into account characteristics of the illumination, discrimination of facets and stars and even processing of the geometry of an object.

[4] `ftp://graphics.cs.msu.ru/projects/PurpleFires/2012-11-28-Segmentation results/manual.html`

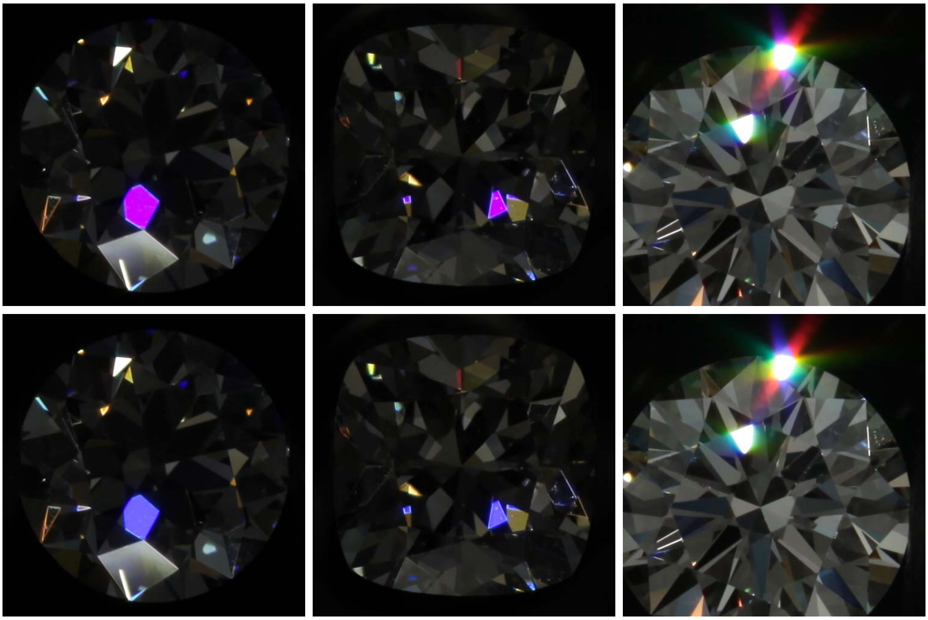

Fig. 9. Examples of the correction made by the proposed algorithm. The source images are shown in the top row. The corrected images are illustrated in the bottom row.

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

References

1. Bruce lindbloom's web site, `http://brucelindbloom.com` (accessed: May 27, 2012)
2. CGAL - computational geometry algorithms library, `http://www.cgal.org` (accessed: May 27, 2012)
3. Decoding raw digital photos in linux, `http://cybercom.net/~dcoffin/dcraw` (accessed: November 27, 2012)
4. Intel integrated performance primitives, `http://software.intel.com/en-us/intel-ipp` (accessed: November 28, 2012)
5. Adams, J., Parulski, K., Spaulding, K.: Color processing in digital cameras. IEEE Micro 18(6), 20–30 (1998)
6. ANSI, I.: 7/2-1993 (ISO 12641). Graphic Technology-Color Reflection Target for Input Scanner Calibration
7. Hong, G., Luo, M., Rhodes, P.: A study of digital camera colorimetric characterisation based on polynomial modelling (2001)
8. Matrosov, M., Ignatenko, A., Sivovolenko, S.: Detection and correction of unnatural purple colors in images of refractive objects taken by digital still camera. In: Graphicon (2012)

9. McCamy, C., Marcus, H., Davidson, J.: A color-rendition chart. J. App. Photog. Eng. 2(3), 95–99 (1976)
10. Newhall, S., Nickerson, D., Judd, D.: Final report of the OSA subcommittee on the spacing of the munsell colors. JOSA 33(7), 385–411 (1943)
11. Park, J., Park, K.: Professional colour communicator-the definitive colour selector. Journal of the Society of Dyers and Colourists 111(3), 56–57 (1995)
12. Smith, T., Guild, J.: The CIE colorimetric standards and their use. Transactions of the Optical Society 33, 73 (1931)
13. Spaulding, K., Vogel, R., Szczepanski, J.: Method and apparatus for color-correcting multi-channel signals of a digital camera, US Patent 5,805,213 (September 8, 1998)