# Real-Time High-Quality Specular Highlight Removal using Efficient Pixel Clustering

Antonio C. S. Souza*, Márcio C. F. Macedo†, Verônica P. Nascimento*, Bruno S. Oliveira‡
*Department of Computer Science, Federal Institute of Bahia, Salvador, BA, Brazil
Emails: {antoniocarlos,veronicapaixao}@ifba.edu.br
†Department of Computer Science, Federal University of Bahia, Salvador, BA, Brazil
Email: marciocfmacedo@gmail.com
‡Computação Brasil, Salvador, BA, Brazil
Email: boliveira@fapex.org.br

***Abstract*—On the basis of the dichromatic reflection model, recent specular highlight removal techniques typically estimate and cluster illumination chromaticity values to separate diffuse and specular reflection components from a single image. While these techniques are able to obtain visually pleasing results, their clustering algorithms suffer from bad initialization or are too costly to be computed in real time. In this paper, we propose a high-quality pixel clustering approach that allows the removal of specular highlights from a single image in real time. We follow previous work and estimate the minimum and maximum chromaticity values for every pixel. Then, we analyze the distribution pattern of those values in a minimum-maximum chromaticity space to propose an efficient pixel clustering approach. Afterwards, we estimate an intensity ratio for each cluster in order to separate diffuse and specular components. Finally, we present optimization strategies to implement our approach efficiently for both CPU and GPU architectures. Experimental results evaluated in the available dataset show that the proposed approach is not only more accurate, but is also two times faster than the state-of-the-art when running solely on the CPU. Running on the GPU, we show that our approach requires $\approx$ 24 milliseconds to remove specular highlights in an image with $3840 \times 2160$ (4k) resolution. That makes our GPU implementation more than one order of magnitude (20 $\times$) faster than the state-of-the-art for 4k resolution images, while providing the desired effect accurately.**

## I. Introduction

Computer vision applications, such as intrinsic image decomposition [1], typically assume the materials present in the scene to be purely Lambertian (*i.e.,* without specularity) and consider the regions with specular reflection as noise or outliers. In this case, if a large region of specular highlight is present in the image, the accuracy of the application may decrease severely [2].

To improve the accuracy of those applications, specular highlight removal may be applied as a pre-processing step in the input image. While most of the existing techniques (*e.g.,* [3]–[5]) still perform this task at non-interactive frame rates, applications such as object tracking [6] require real-time solutions to improve the tracking accuracy through the removal of the specular highlight frame by frame.

To ease the task of specular highlight removal, the dichromatic reflection model [7] is the most common assumption used to guide the separation of diffuse and specular reflection components from a single image. In such a model, the color observed at a pixel can be described by the sum of diffuse and specular reflection components or by the linear combination of diffuse and specular chromaticities. In general, most of the existing accurate techniques (*e.g.,* [5], [8], [9]) convert the color of the input pixels into a chromaticity-based space that allows the separation of diffuse and specular components by means of pixel clustering. While these techniques are able to minimize the presence of specular highlights in the image, most of them make use of clustering strategies that are prone to bad initialization of the cluster seeds or are too expensive in terms of processing time.

In this paper, we present an algorithm for real-time, high-quality specular highlight removal from still images that is more accurate and faster than related work, while providing real-time performance even for high-resolution images. Inspired by the work of Shen and Zheng [8], we present an efficient and accurate algorithm to cluster pixels in a minimum-maximum chromaticity space. Then, we estimate an intensity ratio for each cluster to separate diffuse and specular components of the input image. We further discuss strategies to reduce the processing time of the proposed solution, optimizing the performance obtained for both CPU and GPU architectures.

In this work, our main **contributions** are threefold:

1) An efficient pixel clustering algorithm that provides improved initialization of the cluster seeds and achieves results more accurate than related work;
2) An optimization strategy to make the CPU implementation of the proposed solution two times faster than related work;
3) An implementation of the proposed solution on the GPU, achieving results more than one order of magnitude faster than related work for high-resolution images, while still obtaining high accuracy rates;

## II. Related Work

In this section, we present a review of relevant related work proposed in the field of specular highlight removal from a single image. For an in-depth review of the existing specular highlight removal techniques, we refer the reader to see the survey of Artusi *et al.* [10].

2377-5416/18/$31.00 ©2018 IEEE
DOI 10.1109/SIBGRAPI.2018.00014

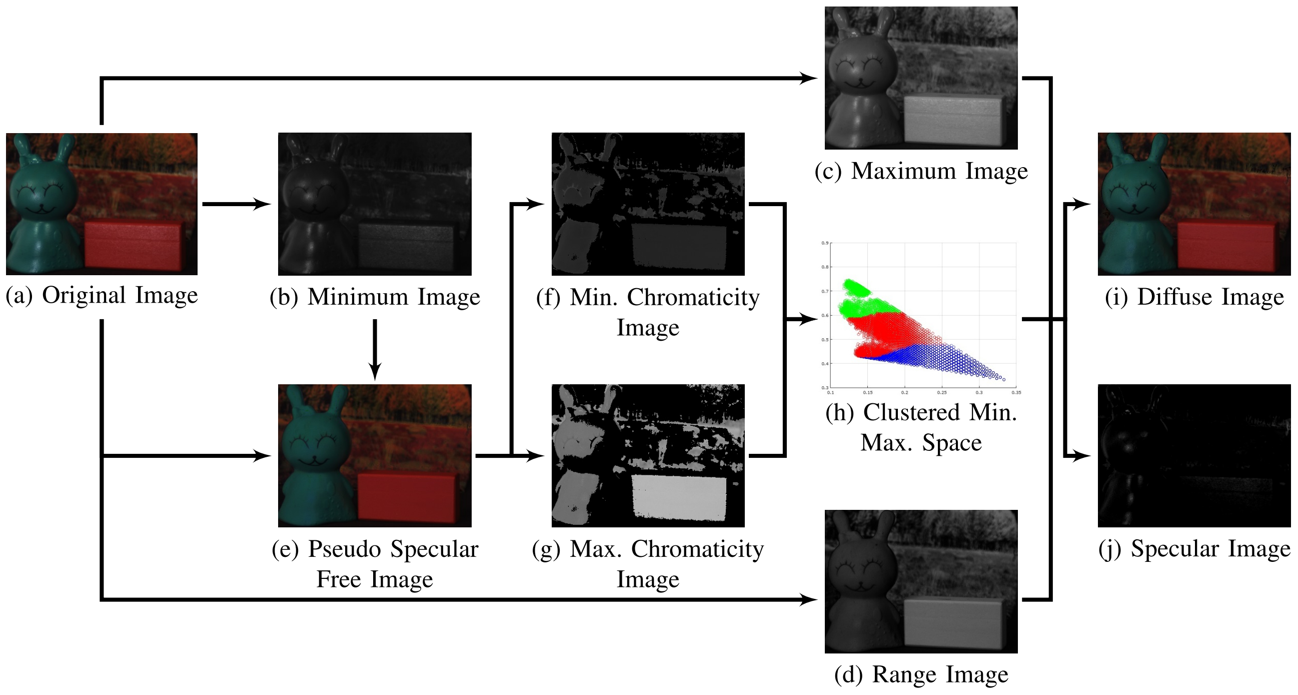

(a) Original Image
(b) Minimum Image
(c) Maximum Image
(d) Range Image
(e) Pseudo Specular Free Image
(f) Min. Chromaticity Image
(g) Max. Chromaticity Image
(h) Clustered Min. Max. Space
(i) Diffuse Image
(j) Specular Image

Fig. 1. An overview of the proposed real-time specular highlight removal algorithm. Given an input image with specular highlights (a), we compute minimum (b), maximum (c) and range (d) values for each pixel. Then, we subtract the input image from the minimum one to estimate the pseudo specular-free image (e). Afterwards, we estimate chromaticity values for each pixel of the pseudo specular-free image and store the minimum (f) and maximum (g) chromaticity channels of relevant pixels into separate images. A minimum-maximum chromaticity space is built and efficiently clustered in only three clusters (h). Next, for each cluster, we estimate a single intensity ratio using maximum and range values. Finally, the three intensity ratios estimated previously are used to compute the real specular-free image (i) through the separation of the specular highlights (j).

Tan *et al.* [11] and Shen *et al.* [12] pioneered the use of a pseudo specular-free image, an approximation of the diffuse-only chromaticity image, to guide an optimization framework and aid the task of specular highlight removal. Because such a framework typically requires more than 5 seconds to remove specular highlights of low-resolution (*e.g.,* 480p) images, its use is inappropriate for real-time applications.

Shen and Cai [13] and Yang *et al.* [14] have adapted the use of connected components labeling and iterative bilateral filtering for specular highlight removal, respectively. Both approaches are almost two orders of magnitude faster than previous work, achieve interactive frame rates, but still demand more than 1 second to remove the specular highlight of high-resolution (*e.g.,* 2160p) images.

Shen and Zheng [8] clustered pixels into a minimum-maximum chromaticity space. Then, for each cluster, the intensity ratio between maximum and range chromaticities was estimated for each pixel and a single intensity ratio was assigned for the entire cluster to separate specular and diffuse pixels accurately.

Rather than relying on the maximum chromaticity-intensity space [11] or the minimum-maximum chromaticity space [8], different color spaces (*e.g.,* Ch-Cv and HSI space [15], a*-b* space [16], hue space [4]) and pixel clustering strategies (*e.g.,* region growing, mean-shift, k-means, affinity propagation) have been used for specular highlight removal. The techniques [15], [17] that use a color segmentation algorithm (*e.g.,* region growing, mean-shift) are prone to noise artifacts and do not work well for textured or multicolored surfaces. The techniques [3]–[5] that use k-means or affinity propagation as a clustering strategy typically require a lot of iterations to converge to an accurate solution, producing frame rates far from real time. Meanwhile, the techniques [8], [9] that are able to achieve high accuracy rates at an interactive processing time may suffer from the bad selection of the initial cluster seeds.

In this work, we take advantage of the specular highlight removal algorithm described by Shen and Zheng [8] to propose an improved and efficient pixel clustering scheme that is more accurate and faster than the original approach. We also present an optimized algorithm to estimate the single intensity ratio of each cluster, which is used to separate diffuse and specular intensities. Finally, we show how the proposed approach can be efficiently implemented on the GPU, achieving, to the best of our knowledge, the best accuracy and processing time results in the literature for the task of specular highlight removal of single images.

## III. Real-Time Specular Highlight Removal

In this section, we introduce our approach for real-time specular highlight removal. An overview of the proposed approach is shown in Figure 1. We first compute the minimum (Figure 1-(b)), maximum (Figure 1-(c)) and range (Figure

1-(d)) values for every pixel. Then, we compute a pseudo specular-free image (Figure 1-(e)) that is used to estimate the minimum (Figure 1-(f)) and maximum (Figure 1-(g)) chromaticity values for every pixel. We analyze the distribution pattern of those values in a minimum-maximum chromaticity space and propose a **novel** clustering scheme that splits the chromaticity space in only three clusters (Figure 1-(h)). Next, for each cluster, we propose a **new** algorithm to estimate a single intensity ratio of maximum and range values per cluster to generate a specular-free image (Figure 1-(i)) in real time.

Let $\mathsf{I}$ be an image where each pixel $x$ in $\mathsf{I}(x)$ = $[\mathsf{I}_\mathrm{r}(x)$, $\mathsf{I}_\mathrm{g}(x)$, $\mathsf{I}_\mathrm{b}(x)]^T$ stores red, green and blue color channels, respectively (Figure 1-(a)). According to the dichromatic reflection model [7], a pixel $\mathsf{I}(x)$ can be represented by the sum of its diffuse $\mathsf{D}(x)$ and specular $\mathsf{S}(x)$ reflection components

$$\mathsf{I}(x) = \mathsf{D}(x) + \mathsf{S}(x), \tag{1}$$

or, alternatively, by the linear combination of diffuse $\Lambda$ and specular $\Gamma$ chromaticity values

$$\mathsf{I}(x) = w_\mathrm{d}(x)\Lambda(x) + w_\mathrm{s}(x)\Gamma, \tag{2}$$

where $w_\mathrm{d}$ and $w_\mathrm{s}$ are weights related to diffuse and specular reflections over the surface geometry, $\Lambda(x) = [\Lambda_\mathrm{r}(x), \Lambda_\mathrm{g}(x), \Lambda_\mathrm{b}(x)]^T$, and $\Gamma$ is typically [5], [8], [9] assumed to be uniform and normalized for the input image $\Gamma = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]^T$, as done in the rest of this paper.

An image with specular highlights typically contain, for the same color surface, pixels with only the diffuse reflection component, and pixels with both diffuse and specular reflection components. Therefore, to remove the specular highlights of an image, we need to define a color space where we are able to cluster and further separate the pure diffuse from the specular pixels. Following related work [8], let us use the concept of intensity ratio for diffuse and specular component separation.

Let $\mathsf{I}^\mathrm{min}$ and $\mathsf{I}^\mathrm{max}$ be single-channel images that store the minimum (Figure 1-(b))

$$\begin{aligned}\mathsf{I}^\mathrm{min}(x) &= \min(\mathsf{I}_\mathrm{r}(x), \mathsf{I}_\mathrm{g}(x), \mathsf{I}_\mathrm{b}(x))\\ &= w_\mathrm{d}(x)\Lambda^\mathrm{min}(x) + w_\mathrm{s}(x)\Gamma,\end{aligned} \tag{3}$$

and the maximum (Figure 1-(c))

$$\begin{aligned}\mathsf{I}^\mathrm{max}(x) &= \max(\mathsf{I}_\mathrm{r}(x), \mathsf{I}_\mathrm{g}(x), \mathsf{I}_\mathrm{b}(x))\\ &= w_\mathrm{d}(x)\Lambda^\mathrm{max}(x) + w_\mathrm{s}(x)\Gamma,\end{aligned} \tag{4}$$

of the red, green and blue intensities per pixel, where $\Lambda^\mathrm{min}(x) = \min(\Lambda_\mathrm{r}(x), \Lambda_\mathrm{g}(x), \Lambda_\mathrm{b}(x))$ and $\Lambda^\mathrm{max}(x) = \max(\Lambda_\mathrm{r}(x), \Lambda_\mathrm{g}(x), \Lambda_\mathrm{b}(x))$.

On the basis of (3) and (4), let $\mathsf{I}^\mathrm{range}$ be a single-channel image that stores the subtraction between $\mathsf{I}^\mathrm{max}$ and $\mathsf{I}^\mathrm{min}$ (Figure 1-(d))

$$\begin{aligned}\mathsf{I}^\mathrm{range}(x) &= \mathsf{I}^\mathrm{max}(x) - \mathsf{I}^\mathrm{min}(x)\\ &= w_\mathrm{d}(x)\Lambda^\mathrm{range}(x).\end{aligned} \tag{5}$$

To prevent $\mathsf{I}^\mathrm{range}(x)$ from being 0, one can add this term by a small value $\epsilon \geq 0$.

From (5), it is easy to see that $\mathsf{I}^\mathrm{range}$ is a specular-free image, since the image contains no longer the specular term $w_\mathrm{s}(x)\Gamma$ of the dichromatic reflection model (2). This effect is also visible in Figure 1-(d).

Now, let us define the intensity ratio $\mathsf{I}^\mathrm{ratio}(x)$ as the ratio between maximum and range values

$$\mathsf{I}^\mathrm{ratio}(x) = \frac{\mathsf{I}^\mathrm{max}(x)}{\mathsf{I}^\mathrm{range}(x)}. \tag{6}$$

For a pixel with pure diffuse reflection, the intensity ratio is clearly defined by the ratio between maximum and range diffuse chromaticities

$$\begin{aligned}\mathsf{I}^\mathrm{ratio}(x) &= \frac{w_d(x)\Lambda^\mathrm{max}}{w_d(x)(\Lambda^\mathrm{max} - \Lambda^\mathrm{min})}\\ &= \frac{\Lambda^\mathrm{max}}{\Lambda^\mathrm{max} - \Lambda^\mathrm{min}}.\end{aligned} \tag{7}$$

However, for a pixel with diffuse and specular reflections, the intensity ratio is

$$\mathsf{I}^\mathrm{ratio}(x) = \frac{w_d(x)\Lambda^\mathrm{max} + w_\mathrm{s}(x)\Gamma}{w_d(x)(\Lambda^\mathrm{max} - \Lambda^\mathrm{min})} \tag{8}$$

From (7) and (8), we can state that, for pixels with the same color surface, or the same diffuse chromaticity, the intensity ratio of the specular pixels is higher than that of purely diffuse pixels. In this sense, the intensity ratio may be used as a metric to separate diffuse and specular pixels efficiently.

In order to use $\mathsf{I}^\mathrm{ratio}$ to separate diffuse and specular pixels, we need to first cluster the pixels with nearly the same diffuse chromaticity. To estimate the diffuse chromaticity of a pixel, let us make use of a pseudo specular-free image $\mathsf{I}^\mathrm{psf}$ (Figure 1-(e)), that can be easily obtained by subtracting the input image $\mathsf{I}$ from $\mathsf{I}^\mathrm{min}$

$$\begin{aligned}\mathsf{I}^\mathrm{psf}(x) &= \mathsf{I}(x) - \mathsf{I}^\mathrm{min}(x)\\ &= w_\mathrm{d}(x)\Lambda^\mathrm{psf}(x).\end{aligned} \tag{9}$$

Similarly to $\mathsf{I}^\mathrm{range}$, $\mathsf{I}^\mathrm{psf}$ does not contain the specular term of the dichromatic reflection model (2). Differently from $\mathsf{I}^\mathrm{range}$, $\mathsf{I}^\mathrm{psf}$ is a three-channel image that resembles the original input image, allowing the estimation of the diffuse chromaticity for red, green and blue intensities.

To further prevent $\mathsf{I}^\mathrm{psf}$ from being too darker than $\mathsf{I}$ and increase the robustness to image noise, we add the mean $\overline{\mathsf{I}^\mathrm{min}}$ of $\mathsf{I}^\mathrm{min}$ to each pixel of $\mathsf{I}^\mathrm{psf}$ [13].

Once with $\mathsf{I}^\mathrm{psf}$, the diffuse chromaticity value for each pixel $\Lambda^\mathrm{psf}(x)$ is estimated as

$$\Lambda^\mathrm{psf}(x) = \frac{\mathsf{I}^\mathrm{psf}(x)}{\mathsf{I}^\mathrm{psf}_\mathrm{r}(x) + \mathsf{I}^\mathrm{psf}_\mathrm{g}(x) + \mathsf{I}^\mathrm{psf}_\mathrm{b}(x)}. \tag{10}$$

As can be seen from (7) and (8), the intensity ratio of a pixel is computed on the basis of minimum and maximum diffuse chromaticity values. Then, we take advantage of $\mathsf{I}^\mathrm{psf}$ (9) and $\Lambda^\mathrm{psf}$ (10) to compute the minimum $\Lambda^\mathrm{psf}_\mathrm{min}(x) = \min(\Lambda^\mathrm{psf}_\mathrm{r}(x), \Lambda^\mathrm{psf}_\mathrm{g}(x), \Lambda^\mathrm{psf}_\mathrm{b}(x))$ (Figure 1-(f)) and maximum $\Lambda^\mathrm{psf}_\mathrm{max}(x) = \max(\Lambda^\mathrm{psf}_\mathrm{r}(x), \Lambda^\mathrm{psf}_\mathrm{g}(x), \Lambda^\mathrm{psf}_\mathrm{b}(x))$ (Figure 1-(g)) pseudo diffuse chromaticity values only for relevant

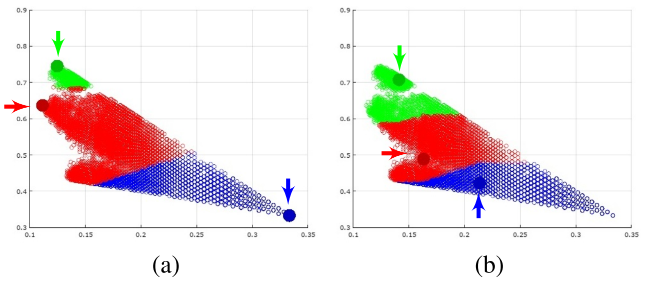


Fig. 2. Our clustering approach for the minimum (x-axis)-maximum (y-axis) chromaticity space. In the first iteration of our algorithm (a), we select three cluster seeds (big circles pointed by arrows) located at the highest minimum (blue circle), highest maximum (green circle), and lowest minimum chromaticity values (red circle). Then, we run two iterations of k-means to associate each pixel to the nearest cluster seed in the Euclidean space, update the position of the cluster seeds according to the centroid of their associated pixels, and associate each pixel on the basis of the updated cluster seeds (b).

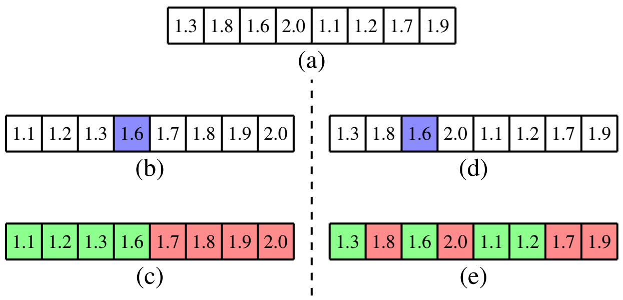


Fig. 3. Given an unordered array of intensity ratios per cluster (a), we can sort such an array (b) and select the median value (blue rectangle in (b)) as the chosen intensity ratio $r$ since this value divides the array equally in diffuse (green rectangles) and specular (red rectangles) intensity ratios (c). An alternative approach is to iteratively select a value (blue rectangle in (d)) from the unordered array that better separates diffuse and specular intensity ratios (e) in the cluster.

pixels that may lie in a region with specular highlights. We determine whether a pixel is relevant to the specular highlight removal as follows

$$\mathsf{I}^{\text{psf}}(x) = \begin{cases} \text{relevant}, & \text{if } \mathsf{I}^{\text{min}}(x) > \overline{\mathsf{I}^{\text{min}}} \\ \text{non-relevant}, & \text{otherwise.} \end{cases} \quad (11)$$

In this sense, background pixels and pixels with an intensity value lower than the average minimum intensity $\overline{\mathsf{I}^{\text{min}}}$ are discarded from further computation since they probably do not contain a specular pixel.

To cluster the pixels with the same diffuse chromaticity, we project the relevant pixels of $\mathsf{I}^{\text{psf}}$ into a minimum-maximum chromaticity space, as shown in Figure 2. Any pixel in this space is projected in the form of a triangle (the proof of this statement can be seen in the Section I of the supplementary document), like the one visible in Figure 2. Rather than selecting the number and the position of the initial cluster seeds randomly, we take advantage of the triangle-like shape where the samples of the minimum-maximum chromaticity space lie within to propose an efficient approach for pixel clustering in this space.

Experimentally, we have seen that, rather than dividing the space into one or two clusters, the separation of the minimum-maximum chromaticity space in three clusters provides an accurate solution to the clustering of relevant pixels. To do so, we first select the pixels whose projections in the minimum-maximum chromaticity space store the highest minimum, highest maximum and lowest chromaticity values as cluster seeds, as shown by the big circles in Figure 2-(a). These pixels are selected as cluster seeds in our algorithm because they can be easily detected from $\Lambda^{\text{psf}}_{\text{min}}$ and $\Lambda^{\text{psf}}_{\text{max}}$, while their projections are vertices of the triangle-like shape formed in the minimum-maximum chromaticity space. Then, we run only two iterations of k-means [18] to associate each relevant pixel of the input image to the nearest cluster seed in the minimum-maximum chromaticity space, to update the position of each cluster seed to be the centroid of its associated pixels projected in the minimum-maximum chromaticity space (see the new positions of the cluster seeds in Figure 2-(b)), and to associate each pixel to the nearest updated cluster seed (Figure 2-(b)). With this algorithm, we are able to remove the randomness to select the initial cluster seeds, providing an improved initialization scheme that is invariant to the orientation of the image, can be easily computed from $\mathsf{I}^{\text{psf}}$, and requires two iterations of k-means to converge to an accurate solution.

After clustering the pixels with nearly the same diffuse chromaticity, we need to select, for each cluster, the intensity ratio $r$ that separates the diffuse pixels from the specular pixels on the basis of $\mathsf{I}^{\text{ratio}}$ (7) and (8) (Figure 3). A clever idea to compute $r$ per cluster is to sort, in an ascending order, the intensity ratios of each pixel associated with the same cluster seed (Figure 3-(b)), and then take the intensity ratio located in the central position of the sorted array as the chosen value $r$ (1.6 in Figure 3-(b)), since it divides the cluster almost equally into diffuse and specular pixels (Figure 3-(c)).

Knowing that, depending on the size of the input array, the sorting step may be too inefficient in terms of processing time, we propose an alternative approach to estimate $r$ without requiring any sorting per cluster. The pseudocode of our algorithm is listed in Algorithm 1. In our proposal, we first compute $r$ as the average intensity ratio for each cluster (Line 6 of Algorithm 1, Figure 3-(d)), and then, we determine the number of pixels in the cluster whose intensity ratio is lower or higher than $r$ (Lines 7, 9, and 10 of Algorithm 1). If $r$ does not divide almost equally the number of pixels into diffuse and specular groups, or, in order words, if one of the groups contain more pixels than $\beta$ fraction of the total number of pixels in that cluster (Lines 11 and 13 of Algorithm 1), we change $r$ by a pre-defined step size $\gamma$ (Lines 12 and 14 of Algorithm 3) and reiterate the algorithm until the maximum number of iterations $\alpha$ has been achieved (Line 8 of Algorithm 1) or the updated $r$ divides diffuse and specular pixels properly (Line 15 of Algorithm 3, Figure 3-(e)). Finally, we assign $\mathsf{I}^{\text{ratio}}(x) = r$ for every pixel $x$ associated with the cluster whose selected intensity ratio is $r$ (Lines 18-22 of Algorithm 3).

**Algorithm 1** Single intensity ratio estimation without array sorting

```
Input: $\mathsf{I}_{\text{max}}$ : an image that stores maximum intensities,
    $\mathsf{I}_{\text{range}}$ : an image that stores range intensities,
    $\alpha$ : maximum number of iterations,
    $\beta$ : a threshold value,
    $\gamma$ : a step value
Output: $\mathsf{I}_{\text{ratio}}$ : an image with selected intensity ratios
1:  procedure ESTIMATERATIO($\mathsf{I}_{\text{max}}$, $\mathsf{I}_{\text{range}}$, $\alpha$, $\beta$, $\gamma$)
2:      for each pixel $x$ do
3:          $\mathsf{I}_{\text{ratio}}(x) \leftarrow \frac{\mathsf{I}_{\text{max}}(x)}{\mathsf{I}_{\text{range}}(x)}$;
4:      end for
5:      for each cluster $c$ do
6:          $r \leftarrow$ average $\mathsf{I}_{\text{ratio}}(x)$ for pixels $x$ projected in $c$;
7:          $n \leftarrow$ count the number of pixels projected in $c$;
8:          while iteration $< \alpha$ do
9:              $l \leftarrow$ count pixels in $c$ whose $\mathsf{I}_{\text{ratio}}(x) \leq r$;
10:             $h \leftarrow n - l$;
11:             if $\frac{l}{n} > \beta$ then
12:                 $r \leftarrow r - \gamma r$;
13:             else if $\frac{h}{n} > \beta$ then
14:                 $r \leftarrow r + \gamma r$;
15:             else break;
16:             end if
17:         end while
18:         if pixel $x$ is projected in $c$ then
19:             $\mathsf{I}_{\text{ratio}}(x) \leftarrow r$;
20:         end if
21:     end for
22:     return $\mathsf{I}_{\text{ratio}}$;
23: end procedure
```

Once the intensity ratio $r$ has been determined per cluster, we can compute the real specular image $\mathsf{S}$ (1) (Figure 1-(j)) as follows. Let us recall from (1) and (2) that $\mathsf{S}(x) = w_{\text{s}}(x)\Gamma$. Now, we can see from (4) and (6) that, for a pixel with pure diffuse reflection, $\mathsf{I}^{\text{max}}(x) = w_{\text{d}}(x)\Lambda^{\text{max}}(x)$ and $\mathsf{I}^{\text{max}}(x) = \mathsf{I}^{\text{ratio}}(x)\mathsf{I}^{\text{range}}(x)$, respectively. Consequently, $w_{\text{d}}(x)\Lambda^{\text{max}}(x) = \mathsf{I}^{\text{ratio}}(x)\mathsf{I}^{\text{range}}(x)$. Then, $\mathsf{S}$ can be computed in terms of the maximum intensity image (4) as

$$\begin{aligned} \mathsf{I}^{\text{max}}(x) &= w_{\text{d}}(x)\Lambda^{\text{max}}(x) + w_{\text{s}}(x)\Gamma \\ w_{\text{s}}(x)\Gamma &= \mathsf{I}^{\text{max}}(x) - w_{\text{d}}(x)\Lambda^{\text{max}}(x) \\ \mathsf{S}(x) &= \mathsf{I}^{\text{max}}(x) - \mathsf{I}^{\text{ratio}}(x)\mathsf{I}^{\text{range}}(x). \end{aligned} \tag{12}$$

From (6), $\mathsf{S}(x)$ would be zero in (12) because $\mathsf{I}^{\text{max}}(x) = \mathsf{I}^{\text{ratio}}(x)\mathsf{I}^{\text{range}}(x)$. However, as listed in Algorithm 1, after the computation of (6), the values of $\mathsf{I}^{\text{ratio}}$ are changed such that a single intensity ratio $r$ is used per cluster, making (6) an approximated equation.

Finally, the diffuse-only image $\mathsf{D}$ (Figure 1-(i)) can be easily computed by the subtraction between the input image and $\mathsf{S}$ (1).

## IV. GPU-Based Specular Highlight Removal

In this section, we describe our GPU implementation of the pipeline depicted in Figure 1. We assume, by default, that each thread inside a kernel performs its computation on the basis of its corresponding pixel in an image. Furthermore, all images handled by our GPU algorithm are created and stored in the pitched memory, optimizing read and write operations by means of coalesced memory access.

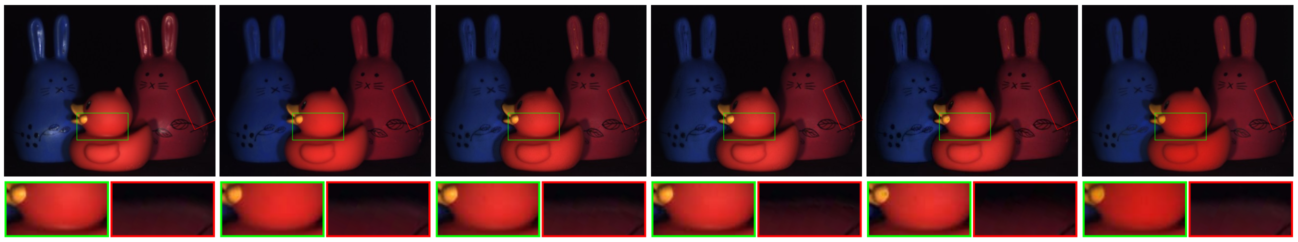
First, we compute $\mathsf{I}_{\text{min}}$ (Figure 1-(b)), $\mathsf{I}_{\text{max}}$ (Figure 1-(c)) and $\mathsf{I}_{\text{range}}$ (Figure 1-(d)) in a single kernel on the GPU. Each thread is responsible for computing minimum (3), maximum (4) and range (5) intensities in parallel.

Next, we estimate $\overline{\mathsf{I}^{\text{min}}}$, the mean of $\mathsf{I}_{\text{min}}$, in order to classify relevant and non-relevant pixels (11). To do so, we run the parallel prefix sum [19] over $\mathsf{I}^{\text{min}}$ in order to sum the minimum intensity values of $\mathsf{I}^{\text{min}}$, and estimate its average minimum intensity.

Afterwards, in another kernel, we compute $\mathsf{I}_{\text{psf}}$ (9) (Figure 1-(e)), $\Lambda^{\text{psf}}$ (10) and store minimum $\Lambda^{\text{psf}}_{\text{min}}$ (Figure 1-(f)) and maximum $\Lambda^{\text{psf}}_{\text{max}}$ (Figure 1-(g)) pseudo diffuse chromaticity values in separate images. For this task, each thread works with its corresponding pixel in parallel.

Before performing the clustering of pixels in the minimum-maximum chromaticity space, we locate the three initial cluster seeds from both $\Lambda^{\text{psf}}_{\text{min}}$ and $\Lambda^{\text{psf}}_{\text{max}}$ on the GPU. We use the optimized scan structure [19] of the parallel prefix sum to recover the pixels with the lowest and highest minimum, and highest maximum pseudo diffuse chromaticity values. As shown in Figure 2-(a), these are the initial cluster seeds of our clustering algorithm. After uploading them in the constant device memory, we associate each relevant pixel with its nearest cluster seed, an operation that is pixel-independent. Therefore, in a separate kernel, each thread projects its corresponding pixel in the minimum-maximum chromaticity space, measures the distance of the projected pixel to the three existing cluster seeds and stores the identifier of the nearest cluster seed to the pixel. For each one of the three clusters, we make use of the scan structure to count, in parallel, the number of pixels associated with each cluster. Next, we run the parallel prefix sum to sum the minimum and maximum diffuse chromaticity values of these pixels and update the centroids of the cluster seeds. After copying these new cluster seed positions to the constant memory, we run another assignment step, in parallel, to associate each relevant pixel with its updated nearest cluster seed (Figure 2-(b)).

As we have shown in Figure 3, there are two ways to compute the intensity ratio $r$ that divides each cluster into diffuse and specular pixels. In both approaches, we first use a thread to compute $\mathsf{I}_{\text{ratio}}$ (6) in parallel. Next, we run the scan operator to count, in parallel, the number of pixels associated with each cluster. For the sorting-based approach, we copy the pixels associated with the same cluster seed into a separate, compact array. Then, we run the parallel radix sort [19] to sort the intensity ratios (Figure 3-(b)) and copy the chosen intensity ratio $r$ for each pixel $x$ of $\mathsf{I}_{\text{ratio}}(x)$ in parallel. The non-sorting-



(a) Original (b) Ground-Truth (c) Ours PNSR 37.5 (d) Shen and Zheng [8] PNSR 37.3 (e) Yang *et al.* [14] PNSR 37.2 (f) Shen and Cai [13] PNSR 34.8

Fig. 4. A visual comparison between our approach (c) and the three best ranked specular removal methods (d-f) for the *Animals* image.

based approach listed in Algorithm 1 can be easily parallelized as well. For each cluster, we run the parallel prefix sum to estimate the average intensity ratio of the pixels associated to the same cluster seed (Lines 6-7 of Algorithm 1). Using the scan operator, we count the pixels whose intensity ratios are lower than $r$ (Line 9 of Algorithm 1). In a single-thread on the CPU, we control the iterations and the update over $r$ (Lines 8, 11-16 of Algorithm 1). Next, in parallel, we copy the chosen intensity ratio $r$ for each pixel $x$ of $\mathsf{I}_{\text{ratio}}(x)$ (Lines 18-19 of Algorithm 1).

Finally, each GPU thread makes use of the intensity ratio $r$ estimated per cluster to compute, in parallel, each pixel of $\mathsf{D}$ (Figure 1-(i)) and $\mathsf{S}$ (Figure 1-(j)) according to the equation (12), that is pixel independent and easily parallelizable.

## V. Results and Discussion

In this section, we evaluate the visual quality and processing time obtained by distinct specular highlight removal techniques proposed in the literature. We restrict our evaluation to the techniques [5], [9] that reported their results on the standard dataset proposed by Shen and Zheng [8], or whose source codes are open and freely available [8], [11]–[14], [20], or were gently shared by their respective authors [4].

### A. Experimental Setup

To measure the processing time of our approach as well as the ones proposed by related work, we have used an NVIDIA GeForce GTX Titan X graphics card and an Intel Core$^{\text{TM}}$ i7-3770K CPU (3.50 GHz), 8GB RAM.

To implement our approach, we used the OpenCV 2.3.1 [21] and CUDA 8.0 [22] for image and parallel processing tasks, respectively. Moreover, we used the optimized implementations of parallel prefix sum and radix sort in the Thrust library[1]. Our final source code is open and available for free[2].

In terms of accuracy, we compare the sorting-based (SB) (Figures 3-(b, c)) and non-sorting-based (NSB) (Figures 3-(d, e)) variants of our approach with respect to related work measuring the peak signal-to-noise ratio (PSNR) over the four images available in the standard specular highlight removal

[1]https://thrust.github.io/

[2]https://github.com/MarcioCerqueira/RealTimeSpecularHighlightRemoval

TABLE I
The accuracy obtained by different specular highlight removal techniques for the four images of a standard dataset [8]. Accuracy is measured in terms of the PSNR metric.

| Method | Masks | Cups | Fruit | Animals |
|---|---|---|---|---|
| Tan *et al.* [11] | 25.6 | 30.2 | 29.6 | 29.9 |
| Shen *et al.* [12] | 32.2 | 37.5 | 38.0 | 34.2 |
| Shen and Cai [13] | 34.0 | 37.6 | 36.9 | 34.8 |
| Q. Yang *et al.* [14] | 32.2 | 38.0 | 37.6 | 37.2 |
| J. Yang *et al.* [15] | 29.8 | 36.4 | 35.6 | – |
| Shen and Zheng [8] | 34.1 | 39.3 | 38.9 | 37.3 |
| Liu *et al.* [4] | 34.5 | 37.6 | 35.1 | 33.4 |
| Suo *et al.* [5] | 34.2 | – | **40.4** | – |
| Ren *et al.* [9] | 34.5 | 38.0 | 37.7 | – |
| **Our SB approach** | **34.8** | **39.6** | 39.3 | **37.4** |
| **Our NSB approach** | **34.9** | **39.5** | 39.4 | **37.5** |

dataset [8]. In terms of performance, we measured the averaged processing time obtained by both CPU and GPU versions of our approach and the CPU implementations of related work for the typical image resolutions of $480p$, $720p$, $1080p$, and $2160p$.

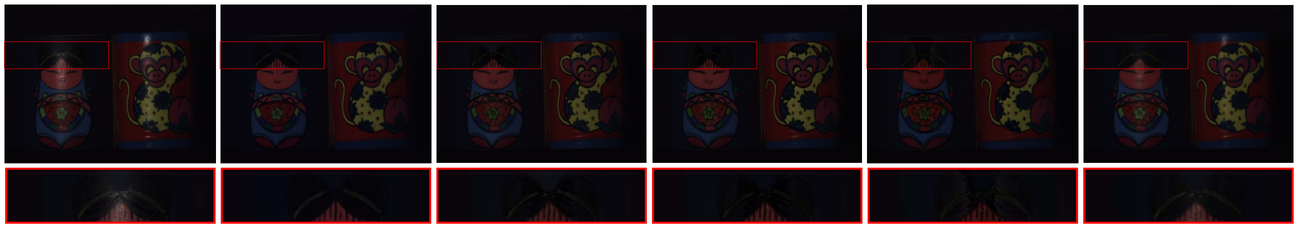
In relation to the parameters listed in Algorithm 1, we have chosen empirically $\alpha = 3$ to keep the real time performance of the algorithm, $\beta = 0.55$ and $\gamma = 0.025$ to enable the algorithm to converge faster to an accurate solution. As we show in the rest of this section, these parameters were sufficient to provide accurate results for our tests.

### B. Accuracy Evaluation

In this section, we evaluate the accuracy obtained by our approach and related work both quantitatively and qualitatively.

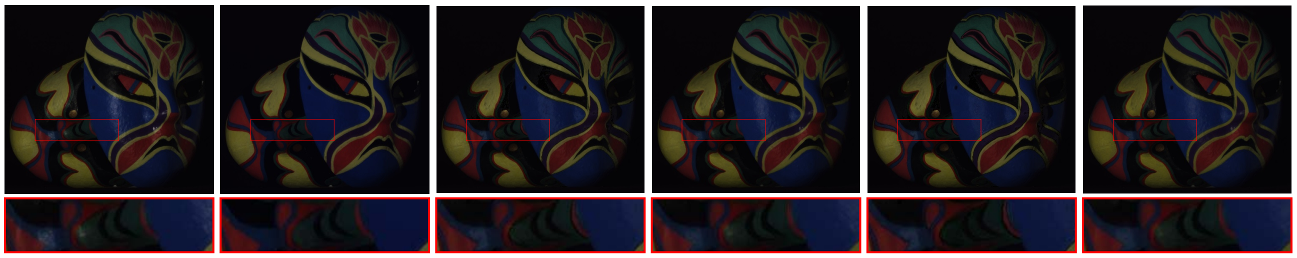
In Table I, we compare the accuracy obtained by our approach and related work for the four images used in the standard dataset [8]. Since we could not obtain the source codes for the techniques of J. Yang *et al.* [23], Suo *et al.* [5] and Ren *et al.* [9], we only show the accuracy results reported in their original papers.

It is visible that the proposed approach is more accurate than the state-of-the-art techniques for three of the four images available in the standard dataset, showing that our clustering scheme is able to remove the specular highlights accurately. It is also visible from Table I that both sorting and non-sorting



(a) Original (b) Ground-Truth (c) Ours PNSR 39.6 (d) Shen and Zheng [8] PNSR 39.3 (e) Yang *et al.* [14] PNSR 38.0 (f) Liu *et al.* [4] PNSR 37.6

Fig. 5. A visual comparison between our approach (c) and three of the best ranked specular removal methods (d-f) for the *Cups* image.



(a) Original (b) Ground-Truth (c) Ours PNSR 34.9 (d) Liu *et al.* [4] PNSR 34.5 (e) Shen and Zheng [8] PNSR 34.1 (f) Shen and Cai [13] PNSR 34.0

Fig. 6. A visual comparison between our approach (c) and three of the best ranked specular removal methods (d-f) for the *Masks* image.

TABLE II
A RANKING OF THE AVERAGED PROCESSING TIME (IN SECONDS) OBTAINED BY DIFFERENT SPECULAR HIGLIGHT REMOVAL TECHNIQUES FOR IMAGES WITH DIFFERENT RESOLUTIONS. SOME RESULTS FOR THE APPROACH OF LIU *et al.* [4] ARE NOT LISTED BELOW BECAUSE THEY RUN OUT OF MEMORY FOR HIGH-RESOLUTION IMAGES.

| | **Image Resolution (s)** | | | |
|---|---|---|---|---|
| **Method** | **480p** | **720p** | **1080p** | **2160p** |
| Liu *et al.* [4] | 70.61 | 616.12 | – | – |
| Tan *et al.* [11] | 8.72 | 29.68 | 71.31 | 360.79 |
| Shen *et al.* [12] | 7.96 | 26.04 | 70.68 | 267.55 |
| Q. Yang *et al.* [14] | 0.11 | 0.29 | 0.63 | 2.48 |
| Shen and Cai [13] | 0.055 | 0.15 | 0.34 | 1.44 |
| Shen and Zheng [8] | 0.023 | 0.066 | 0.14 | 0.54 |
| **Our CPU-SB approach** | **0.017** | **0.046** | **0.10** | **0.47** |
| **Our CPU-NSB approach** | **0.011** | **0.030** | **0.068** | **0.26** |
| **Our GPU-NSB approach** | **0.015** | **0.017** | **0.020** | **0.030** |
| **Our GPU-SB approach** | **0.013** | **0.015** | **0.017** | **0.024** |

variants of our approach provide nearly the same accuracy, being able to obtain better accuracy than related work.

In Figures 4, 5, and 6, we provide a visual comparison between our approach and three of the most accurate related work, except for the techniques whose source codes were not openly available [5], [9], [15]. We refer the reader to see the Section II of the supplementary document for a more complete visual analysis of the results. The techniques proposed by Tan *et al.* [11], Shen *et al.* [12] and Shen and Cai [13] reduce the specular highlights found in the original image, but they also change the color intensity for diffuse-only regions of the image. This artifact is mainly visible in the green closeup of Figure 4-(f), where the red color of the duck was darkened. That happens because the iterative framework of Tan *et al.* [11] works on the basis of a specular-free image that is not consistent in terms of both color and geometry profiles, while the other works [12], [13] are susceptible to noise artifacts. The works of Yang *et al.* [14] and Shen and Zheng [8] tend to reduce the color intensity of bright diffuse regions located near the border of the objects. This fact can be seen in the red closeups of Figures 4-(d, e). Also, techniques such as Shen and Cai [13], Yang *et al.* [14] and Liu *et al.* [4] cannot accurately reduce the specular highlight found in textured objects, like the ones shown in Figures 5-(e, f) and Figure 6-(d). In this sense, our approach is able to better handle the specular highlight removal for textured objects (Figures 5-(c) and 6-(c)) and preserve the bright intensities found in diffuse-only regions of the image (Figures 4-(c)).

## C. Processing Time Evaluation

In Table II, we provide a processing time comparison between our approach and related work for varying image resolution. The work of Liu *et al.* [4] is the slowest one because their algorithm uses costly optimization steps to remove specular highlights. The approaches of Tan *et al.* [11] and Shen *et al.* [12] use iterative algorithms that are not scalable for large image resolutions, providing performance far from real time. The algorithm proposed by Q. Yang *et al.* [14] provides scalability for the specular highlight removal, but needs between two and four iterations of bilateral filtering to converge to a good solution, demanding more than 1 second to remove the specular highlight for images with $2160p$ resolution. Shen and Cai [13] make use of a connected components labeling algorithm to detect the largest region of highlight in the input image, dilate such a region to detect surrounding diffuse pixels and solve a least squares problem to remove the specular highlight. Such a simple approach achieves real-time performance for low-resolution images, but is not scalable for high-resolution images. As shown in Table II, the fastest techniques for specular highlight removal are the ones based on the clustering of the minimum-maximum chromaticity space, namely Shen and Zheng [8] and our approach. However, we show that, through the use of a non-sorting-based approach to estimate the single intensity ratio that separates diffuse and specular pixels per cluster, we are able to reduce by a half the processing time obtained by the fastest related work [8], obtaining the fastest and most accurate CPU-based algorithm for specular highlight removal. However, as can be seen in Table II, our CPU-based algorithm does not run in real-time for $1080p$ and $2160p$ image resolutions. In this sense, we show that our GPU-based implementation provides scalability with respect to the image resolution, being at least $20\times$ faster than the work of Shen and Zheng [8] and requiring only 24 milliseconds to remove the specular highlights of a $2160p$ image, achieving real-time performance for both low- and high-resolution images.

## VI. Conclusion and Future Work

In this paper, we have presented a real-time, high-quality algorithm for specular highlight removal of single images. Using a simple, yet efficient pixel clustering algorithm coupled with an optimized approach to separate diffuse from specular pixels, we have shown that our approach is able to achieve processing time two times faster than the state-of-the-art, while providing the most accurate results for three of the four images available in the existing dataset for specular highlight removal. Moreover, with our GPU-based implementation, we are able to provide real-time performance for the specular highlight removal, even for $2160p$ image resolutions. In this sense, we could enable the use of accurate specular highlight removal for real-time applications such as augmented reality and real-time image processing.

We believe that the field of specular highlight removal still needs the development of a larger specular highlight removal dataset, to enable a better evaluation of the generality of the techniques already proposed in the literature, meanwhile favoring the use of deep learning strategies to learn the best features able to provide high-quality specular highlight removal. Also, we intend to further investigate whether the use of alternative luminance-chrominance color spaces may enable us to achieve better accuracy results.

## Acknowledgment

We would like to thank the authors of related work [4], [8], [11], [14] for sharing their source codes, and NVIDIA for providing the GeForce GTX Titan X through the GPU Education Center program. Márcio C. F. Macedo is financially supported by the Coordenação de Aperfeiçoamento de Pessoal do Nível Superior (CAPES).

## References

[1] J. T. Barron and J. Malik, "Shape, Illumination, and Reflectance from Shading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1670–1687, Aug 2015.

[2] Y. Han, J. Y. Lee, and I. S. Kweon, "High Quality Shape from a Single RGB-D Image under Uncalibrated Natural Illumination," in *Proceedings of the IEEE ICCV*, Dec 2013, pp. 1617–1624.

[3] H. Kim, H. Jin, S. Hadap, and I. Kweon, "Specular Reflection Separation Using Dark Channel Prior," in *Proceedings of the IEEE CVPR*, June 2013, pp. 1460–1467.

[4] Y. Liu, Z. Yuan, N. Zheng, and Y. Wu, "Saturation-preserving specular reflection separation," in *Proceedings of the IEEE CVPR*, June 2015, pp. 3725–3733.

[5] J. Suo, D. An, X. Ji, H. Wang, and Q. Dai, "Fast and High Quality Highlight Removal From a Single Image," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5441–5454, Nov 2016.

[6] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake, "MIS-SLAM: Real-time Large Scale Dense Deformable SLAM System in Minimal Invasive Surgery Based on Heterogeneous Computing," in *arXiv preprint*. arXiv:1803.02009, 2018.

[7] S. A. Shafer, "Using color to separate reflection components," *Color Research & Application*, vol. 10, no. 4, pp. 210–218, 1985.

[8] H.-L. Shen and Z.-H. Zheng, "Real-time highlight removal using intensity ratio," *Appl. Opt.*, vol. 52, no. 19, pp. 4483–4493, Jul 2013.

[9] W. Ren, J. Tian, and Y. Tang, "Specular Reflection Separation With Color-Lines Constraint," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2327–2337, May 2017.

[10] A. Artusi, F. Banterle, and D. Chetverikov, "A Survey of Specularity Removal Methods," *Computer Graphics Forum*, vol. 30, no. 8, pp. 2208–2230.

[11] R. T. Tan and K. Ikeuchi, "Separating reflection components of textured surfaces using a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 178–193, Feb 2005.

[12] H.-L. Shen, H.-G. Zhang, S.-J. Shao, and J. H. Xin, "Chromaticity-based Separation of Reflection Components in a Single Image," *Pattern Recogn.*, vol. 41, no. 8, pp. 2461–2469, Aug. 2008.

[13] H.-L. Shen and Q.-Y. Cai, "Simple and efficient method for specularity removal in an image," *Appl. Opt.*, vol. 48, no. 14, pp. 2711–2719, May 2009.

[14] Q. Yang, J. Tang, and N. Ahuja, "Efficient and Robust Specular Highlight Removal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1304–1311, June 2015.

[15] J. Yang, Z. Cai, L. Wen, Z. Lei, G. Guo, and S. Z. Li, "A new projection space for separation of specular-diffuse reflection components in color images," in *Proceedings of the ACCV*. Berlin, Heidelberg: Springer, 2013, pp. 418–429.

[16] T. Nguyen, Q. Vo, S. Kim, H. Yang, and G. Lee, "A novel and effective method for specular detection and removal by tensor voting," in *Proceedings of the IEEE ICIP*, Oct 2014, pp. 1061–1065.

[17] E. Angelopoulou, "Specular Highlight Detection Based on the Fresnel Reflection Coefficient," in *Proceedings of the IEEE ICCV*, Oct 2007, pp. 1–8.

[18] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[19] M. Harris, S. Sengupta, and J. D. Owens, "Parallel prefix sum (scan) with CUDA," *GPU gems*, vol. 3, no. 39, pp. 851–876, 2007.

[20] R. T. Tan and K. Ikeuchi, "Reflection components decomposition of textured surfaces using linear basis functions," in *Proceedings of the IEEE CVPR*, vol. 1, June 2005, pp. 125–131.

[21] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*, 2nd ed. O'Reilly Media, Inc., 2013.

[22] D. B. Kirk and W.-m. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013.

[23] J. Yang, L. Liu, and S. Z. Li, "Separating Specular and Diffuse Reflection Components in the HSI Color Space," in *Proceedings of the IEEE ICCVW*, Dec 2013, pp. 891–898.