International Journal Of Advance Research, Ideas And Innovations In Technology

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 5)
Available online at: www.ijariit.com

Elimination of specular reflection

Samruddhi Patil
samu226741@gmail.com
Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra

Chaya Jadhav
chaya123jadhav@gmail.com
Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra

ABSTRACT

Detection of highlights is a prominent issue in computer vision, graphics and image processing. Applications that require object properties' measurement or rendering are affected by specular reflection since the models assume matte diffusing surfaces most of the time. Hence, detection, and sometimes removal, of specular reflection (highlights) in an image may be critical. Proposed work takes the help of global color information of an image to effectively recover specular and diffuse reflection. We take the help of a dichromatic reflection model to get insights into specular and diffuse reflection. To detect illumination chromaticity (specular reflection regions) we have effectively used a k-means algorithm. Pixels in an image are spread over the entire image and their distances to the illumination chromaticity give the number of specular reflection components. With such non-local overall information from these color lines, our method can effectively separate specular and diffuse reflection components using the repaint() method. We have taken the help of the Imglib2 library.

Keywords— *Diffuse reflection, Specular reflection, Illumination chromaticity, Global color information*

1. INTRODUCTION

One of the main problems in computer graphics, vision and image processing is the detection and removal of highlights. The earlier methods of reflection removal often disunite specular reflection from an image by making use of the patch-based approach. Because of the absence of global information, these methods can't entirely distinguish the specular component of an image and tend to degrade prominent textures in image. Specular reflection negatively affects the quality of visuals and cheapens the performance of many algorithms like image segmentation, visual tracking, detection of objects, color constancy, etc.

Thus, it is important to separate specular and diffuse reflection for computer applications. In case of a complex textured image, segmenting an image into color patches is quite difficult, and the recovered image might not be optimally visual. One of the reasons behind such a failure can be the difficulty of cluster image pixels. Wrongly clustered pixels of image leads to poor image textures. Even though the clusters are nicely formed, spreading diffuse chromaticity within a small patch is not effective to separate specular reflection. Also, patch-based methods can't search exact diffuse chromaticities for obtaining specular reflection because they lack global information. The result leads to blocky separation of specular objects.

The proposed work observes that each pixel of an image lies along a color line in the normalized RGB space. Pixels along the same color line spread over the entire image and their distances to the illumination chromaticity determine the amount of specular reflection. With this non-local information, a method to effectively separate and remove specular reflection for color images is provided. By clustering image pixels properly, an approach to correctively calculates the illumination chromaticity of a highlighted image is given. In order to robustly cluster image pixels k means algorithm is used. Then the phenomenon of repaint method is applied to remove specular reflection.

Pixels that are far away from illumination chromaticity are diffuse chromaticities of the images, whereas the ones near the illumination chromaticity are highlight pixels. According to highlight pixel's distance to the illumination chromaticity, specular reflection components are calculated and specular reflection is separated with the help of matting coefficient. The experimental results on natural images show that the proposed method is accurate and robust under known scene illumination chromaticity.

2. LITERATURE SURVEY

D. Berman, S. Avidan et al. [1] have presented an algorithm based on a non-local statistic. Their algorithm makes use of haze lines that recover the distance map and the haze-free image. The basis of these haze lines is that colors in a haze-free image produce tight clusters within RGB space and the pixels in these clusters are non-local. These pixels are actually situated at various distances from camera. These deviated distances transform into various transmission coefficients, resulting in a line in RGB known as haze line. The algorithm is not trained and performs well on a wide variety of images. However the algorithm fails in scenes where air-light is significantly brighter than the scene.

Q. Yang, J. Tang [2] have treated specular pixels as noise. They have utilized the fact that the maximum fraction of diffuse color components in local patches of original color image change smoothly. The highlights are removed using the idea of image denoising. A bilateral filter which is a low pass filter is used to smoothen out the maximal fraction of the color component in order to remove noise made by specular pixels. The performance of their method highly depends on the approximate maximal diffuse chromaticity.

R. Fattal [3] has described a procedure for dehazing of images. One dimensional distribution of pixels of small image patches known as color lines helps in deriving a local model explaining the color lines in the context of hazy scenes. Those haze lines recover the scene transmission based on the lines' offset from the origin. Markov random field model is used to obtain complete transmission maps even when the estimates are noisy and scattered. The drawback of their procedure is that it produces erroneous results when atmospheric light is very close to the sky color in images.

H. L. Shen, Z.H. Zheng [4] have computed intensity ratios for diffuse pixels to obtain specular fractions of image pixels. Intensity ratios between range and maximum values of diffuse pixels do not depend on the geometry of surface. In case of textured surfaces, image pixels are clustered by building a pseudo chromaticity space to obtain intensity ratio for every cluster. The method processes in a pixel-wise manner without identifying specular pixels and without interacting locally in any way. An issue observed here is that the algorithm doesn't work best on dark and achromatic surfaces.

I. Omar, M. Werman [5] have introduced 'color lines'. The color line is basically a representation of color that is image specific. Color lines are robust to color distortion. The proposed method unriddles the problem of deciding whether two image pixels are of same color in real world. The color lines framework of an image is a listing of lines representing image's colors. It also gives a metric for calculating distance between each pixel and every color line. Each color line represents an elongated cluster in RGB histogram. To compute color lines they have constructed histogram slices. For each slice they have located local maxima point and then combined maxima points from neighboring slices to form color lines. Only color information can be used while applying this approach in real-time and not the texture/edge/spatial information.

S. Shafer [6] have proposed an algorithm based upon a physical model of reflection, according to which there are two types of reflection (interface and body reflection). Each type can be decomposed into a spectral distribution and a geometric scale factor. The methodology makes use of properties of spectral projection into color space for deriving a radical framework of pixel-value color distribution. The thing to be noted here is that the noise in the measuring pixel values must be small enough for the plane and parallelogram fitting to be well.

R. T. Tan, K. Nishino, K. Ikeuchi [7] have proposed a separation method for specular and diffuse pixels which is based on pixels' distribution in maximum chromaticity-intensity space. Their method is accurate in separating the reflection when only the diffuse chromaticity of the normalized image is given.

R. Bajcsy, W. L. Sang, A. Leonardis [8] have transformed color pixels from RGB space to S space and examined color fluctuations on objects in terms of hue, brightness, and saturation which are defined in the S space. But the challenge this algorithm faces is that their algorithm is only effective for dielectric surfaces of homogeneous color under single-color scene illumination. So the advancements should be done to resolve this issue.

S. P. Mallick, T. E. Zickler, D. J. Kriegman and P. N. Belhumeur [9] have proposed a color space SUV, which is a rotation of RGB space, to recover the specular and diffuse factor of an image. This method works well for uniform surfaces when the noise is less. But textured surfaces offend the homogeneous assumption.

P. Tan, S. Lin, L. Quan and H. Shum [10] presented highlight removal method within a single image incorporating illumination based constraints into image inpainting. The performance of their method is not optimal for surfaces that are textured and rough which require segmentation of the surface into different diffuse colors.

3. SYSTEM ARCHITECTURE

3.1 Problem Statement

The task is to design and implement a system that can detect, differentiate and separate specular and diffuse reflection of an image.

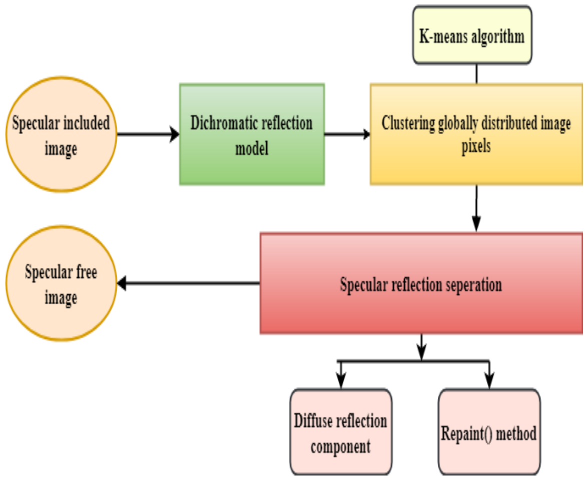

Fig. 1: Proposed System Architecture

The main contributions of this work are presented as follow:

- Every image pixel in a color image lies along a color line in the normalized RGB space. Each color line represents a different diffuse chromaticity. These color lines intersect at one point which is 'illumination chromaticity'. This is helpful for separating specular reection.
- Pixels along the same color line spread over the entire image and their distances to the illumination chromaticity reect the area and amount of specular reection.
- We make use of a k-means clustering algorithm to detect specular regions in the image.
- Then we remove specular pixels by applying the repaint method on the affected area.
- The proposed method separates the specular and diffuse reection of an image and it is suitable for real-time applications.

4. PROPOSED SYSTEM METHODOLOGY

4.1 Dichromatic reflection model

Basically, colors of an image are well approximated by a few hundred distinct colors that form tight clusters in RGB space. The key observation is that pixels in a given cluster are non-local. That is they are spread over entire image place. They are located at different distances. This indicates that pixels of an image can be modeled bylines in RGB space that pass through illumination chromaticity, i.e. illumination chromaticity is obtained when different color lines intersect each other at one point. In other words, every pixel lies along a color line in normalized RGB space. Pixels along a color line come from objects that have similar diffuse chromaticity, located at different distances. These distances from illumination chromaticity react the amount of specular reaction components.

According to dichromatic reection model, RGB color

$$I(y) = [I_r(y), I_g(y), I_b(y)]$$

at a pixel y in an image is a linear combination of diffuse reflection $I_D(y)$ and specular reflection $I_S(y)$. Due to their reflective characteristics, they can be further represented by diffuse chromaticity,

$$\Lambda(y)=[\Lambda_r,\Lambda_g,\Lambda_b]$$

and illumination or specular chromaticity,

$$\Gamma(y) = [\Gamma_r,\Gamma_g,\Gamma_b].$$

So we get,

$$I(y)=I_S(y)+I_D(y)=m_D(y)\Lambda(y)+m_S(y)\Gamma(y)$$

m_D and m_S are the coefficients or surface-geometry related weights for diffuse and specular reflection respectively. For a uniform color surface, diffuse chromaticity values of image pixels are identical. But for a non-uniform or textured surface, they are different. Pixels that are far away from illumination chromaticity are diffuse pixels, whereas the ones near the illumination chromaticity are highlight pixels. According to highlight pixel's distance from illumination chromaticity, we can calculate their specular reection components.

4.2 Clusters Formation

Clustering is performed to distinguish a dataset into a number of groups. We have used k-means clustering for detecting regions with illumination chromaticities i.e. regions with specular reflection in the image. K-means clustering divides a collection of data items into k groups of those data items. After successful pixel calculation, acquired pixels'dataset is classified into 4 disjoint clusters. The algorithm constitutes of two individual stages. Firstly it calculates 4 centroids and in the later stage it assigns each calculated pixel to the cluster that has nearest centroid from the respective pixel. Among the various methods that calculate the distance of the nearest centroid, we have used Euclidean distance. Once the initial grouping of pixels is done, the new centroid of each cluster is recalculated and with the help of this new centroid, we find a new Euclidean distance between each center and each pixel. Pixels are grouped into the cluster having minimum Euclidean distance. Each cluster (Red, Green, Blue, and Alpha) is defined by its member pixels and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the pixels in that cluster is minimized. Hence we have used K-means algorithm as it's an iterative algorithm that minimizes the sum of distances from each pixel to its cluster centroid, over all clusters.

4.3 Algorithm

Input: Buffer Image B, Number of cluster k
Output: Clustered buffer image

Step 1: Calculate total pixels P= [M*N]
Step 2: Generate cluster set R[], G[], B[], A[] and threshold closest
Step 3: for each (pix in P)
Step 4: if ($(pix >> 24)$ & $0xff$)
A.add ← pix
Step 5: if ($(pix >> 16)$ & $0xff$)
R.add ← pix
Step 6: if ($(pix >> 8)$ & $0xff$)
G.add ← pix
Step 7: if (pix & $0xff$)
R.add ← pix
Step 8: *calculate* distance
$weight = ||([RGB].pixelval) - (mean)||$
Step 9: Normalized each pixel using current distance weight
if ($weight \leq closest$)
Temppixelset[] ← pixel[weight]
Step 10: read all Temppixelset and generate final cluster image using below formula

$$clusterset = \sum_{k=0}^{n}\{(Temppixel.pixelval) \ll 24 \,||16||8||0 \ldots.\}$$

Step 11: $return\ clusterset$ as buffered image

4.4 Separating specular reflection

For eliminating the detected specular region We have used repaint () method. It is an asynchronous method. The repaint () method requests an erase and redraw (update) after a small time delay. So by using repaint () we have erased previous specular reflection region from one place and the new paint has updated that region. The repaint () method cannot be overridden. It controls the update () -> paint () cycle. The repaint () method is invoked to do a drawing. There are four versions of this method. We have used the one with no arguments.

Repaint () =>update () => (usually calls) =>paint ()

We call this repaint () method to get specular reection components to repaint themselves. We have used this repaint () method because we wanted to change the look of the specular components (changing its color) and not its size. Repaint () does not invoke paint () directly. It schedules a call to an intermediate method, update (). Finally update () calls paint () method. The reason for this complexity is Java's support for concurrent programming. It does this using threads. The advantage of using repaint () over paint () is that the paint () method is called automatically by the environment that contains the applet whenever the applet window needs to be redrawn. This happens when the component is first displayed, but it can happen again if the user minimizes the window that displays the component and then restores it or if the user moves another window over it and then move that window out of the way. In addition to these implicit calls to the paint () method by the environment, one can also call the paint () method explicitly whenever the applet window needs to be redrawn, using the repaint () method. The repaint () method causes the AWT runtime system to execute the update () method of the Component class which clears the window with the background color of the applet and then calls the paint () method.

5. MATHEMATICAL MODEL

Set ={inpimage,Process,O_image}
Inpimage is the input image while O_image is final specular removed image

Imgpixel=$\sum_{i=0,j=0}^{n} p[i]$ [T]
{x,D,S,I_x}→imgpixel[I_r,I_g, I_b]
Fset={illumination chromaticity, cluster [i…n]}
Cset={cluster[i]….cluster[n]}

Find chromaticity
$D(x)=\{[I_{C_x}]\ldots nvector[x]\}$
$I(x)=\{clusterset\ of\ I_c\ image\}$
$O\_image=\{I(x),D(x)\}$

Success condition
if(inpimage!null&&O_image!=null)

Failure condition
If(D(x)&&I(x)=null&&O_image)

6. RESULTS AND DISCUSSION

Fig. 2: Input Image with specular reflection

Fig. 3: Detected specular regions

Fig. 4: Removed specular region

6.1 Methodology of Evaluation

We have used JAVA 1.7 and MYSQL for the implementation and run it on Pentium IV 2.7 GHz with an i3 processor machine with 2 GB RAM. Windows 7 is used as an operating system.

6.2 Metrics of Evaluation

We evaluated our approach with respect to parameters such as PSNR(Peak Signal to Noise Ratio), MSE (Mean Square Error) and SSIM (Structural Similarity Measurement Index). The error (0.45) is significantly low resulting in a high PSNR value(31.18).

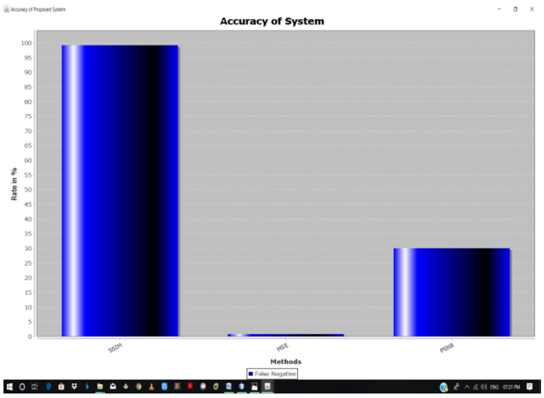

Fig. 5: Evaluation graph

```
run:
fileLength = 78947
Region Detected
paint finish
Execution done
MSE percent: 0.8161342845243567
PSNR = 30.050309966931685
SSIM :99.18386571547565
```

Fig. 6: Evaluated result values

7. CONCLUSION

We have proposed an approach for separating interface and body reflections. Each image pixel lies on a color line in normalized RGB and different color lines (different diffuse chromaticities) intersect at illumination chromaticity. Image pixels are clustered effectively to obtain color lines that captivate global chromaticity knowledge. We then calculate each pixel's distance to illumination chromaticity to separate specular reflection in a pixel-wise manner.

8. ACKNOWLEDGEMENT

I express my gratitude to my mentor Mrs. Chaya Jadhav for her esteemed guidance and encouragement. She always motivates me to succeed further. I am thankful for her guidance and comments during the implementation part of this work. I have learned many things under her leadership.

9. REFERENCES

[1] Weihong Ren, Jiandong Tian, "Specular Reflection Separation with Color-Lines Constraint," IEEE Transactions on Image Processing,2017.
[2] D. Berman, S. Avidan et al., "Non-local image dehazing," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1674–1682.
[3] Q. Yang, J. Tang, and N. Ahuja, "Efficient and robust specular highlight removal," Pattern Analysis Machine Intelligence IEEE Transactions on, vol. 37, no. 6, pp. 1304–11, 2015.

[4] R. Fattal, "Dehazing using color-lines," Acm Transactions on Graphics, vol. 34, no. 1, pp. 1–14, 2014.

[5] H. L. Shen and Z. H. Zheng, "Real-time highlight removal using intensity ratio," Applied Optics, vol. 52, no. 19, pp. 4483–4493, 2013.

[6] I. Omer and M. Werman, "Color Lines: Image Specific Color Representation," IEEE Transactions on Image Processing, CVPR.2004.1315267

[7] S. A. Shafer, "Using color to separate reflection components," Color Research Application, vol. 10, no. 4, p. 210C218, 1985.G. J. Klinker, S. A. Shafer, and T. Kanade, The Measurement of Highlights in Color Images. Jones and Bartlett Publishers, Inc., 1992.

[8] R. T. Tan, K. Nishino, and K. Ikeuchi, "Separating reflection components based on chromaticity and noise analysis," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 10, pp. 1373–1379, 2004.

[9] R. Bajcsy, W. L. Sang, and A. Leonardis, "Detection of diffuse and specular interface reflections and inter-reflections by color image segmentation," International Journal of Computer Vision, vol. 17, no. 3, pp. 241–272, 1996.

[10] S. P. Mallick, T. E. Zickler, D. J. Kriegman, and P. N. Belhumeur, "Beyond lambert: Reconstructing specular surfaces using color," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2. Ieee, 2005, pp. 619–626.

[11] P. Tan, S. Lin, L. Quan, and H. Shum, "Highlight removal by illumination-constrained in painting," in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003, pp. 164–169.

[12] S. W. Lee and R. Bajcsy, "Detection of specularity using color and multiple views," in Computer Vision - ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19-22, 1992, Proceedings, 1992, pp. 99–114.

[13] S. Lin, Y. Li, S. B. Kang, X. Tong, and H. Y. Shum, "Diffuse-specular separation and depth recovery from image sequences," in European Conference on Computer Vision, 2002, pp. 210–224.

[14] S. P. Mallick, T. Zickler, P. N. Belhumeur, and D. J. Kriegman, "Specularity removal in images and videos: A PDE approach," in Computer Vision - ECCV 2006, European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, 2006, pp. 550–563.

[15] S. W. Lee and R. Bajcsy, "Detection of specularity using color and multiple views," in Computer Vision - ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19-22, 1992, Proceedings, 1992, pp. 99–114.

[16] S. Lin, Y. Li, S. B. Kang, X. Tong, and H. Y. Shum, "Diffuse-specular separation and depth recovery from image sequences," in European Conference on Computer Vision, 2002, pp. 210–224.

[17] S. P. Mallick, T. Zickler, P. N. Belhumeur, and D. J. Kriegman, "Specularity removal in images and videos: A PDE approach," in Computer Vision - ECCV 2006, European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, 2006, pp. 550–563.

[18] K. J. Yoon, Y. Choi, and I. S. Kweon, "Fast separation of reflection components using a specularity-invariant image representation," in IEEE

[19] International Conference on Image Processing, 2006, pp. 973–976.

[20] H. L. Shen, H. G. Zhang, S. J. Shao, and J. H. Xin, "Chromaticitybased separation of reflection components in a single image," Pattern Recognition, vol. 41, no. 8, pp. 2461–2469, 2008.

[21] Y. He, N. Khanna, C. J. Boushey, and E. J. Delp, "Specular highlight removal for image-based dietary assessment," in IEEE International Conference on Multimedia and Expo Workshops, 2012, pp. 424–428

