Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2010, Article ID 837237, 22 pages
doi:10.1155/2010/837237

Research Article

Perceptually Motivated Automatic Color Contrast Enhancement Based on Color Constancy Estimation

Anustup Choudhury and Gérard Medioni

Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA

Correspondence should be addressed to Gérard Medioni, medioni@usc.edu

Received 30 March 2010; Revised 20 August 2010; Accepted 1 October 2010

Academic Editor: Zhou Wang

We address the problem of contrast enhancement for color images. Our method to enhance images is inspired from the retinex theory. We try to estimate the illumination and separate it from the reflectance component of an image. We use denoising techniques to estimate the illumination and while doing so achieve color constancy. We enhance only the illumination component of the image. The parameters for enhancement are estimated automatically. This enhanced illumination is then multiplied with the reflectance to obtain enhanced images with better contrast. We provide validation of our color constancy approach and show performance better than state-of-the-art approaches. We also show "visually better" results while comparing our enhancement results with those from other enhancement techniques and from commercial software packages. We perform statistical analysis of our results and quantitatively show that our approach produces effective image enhancement. This is validated by ratings from human observers.

1. Introduction

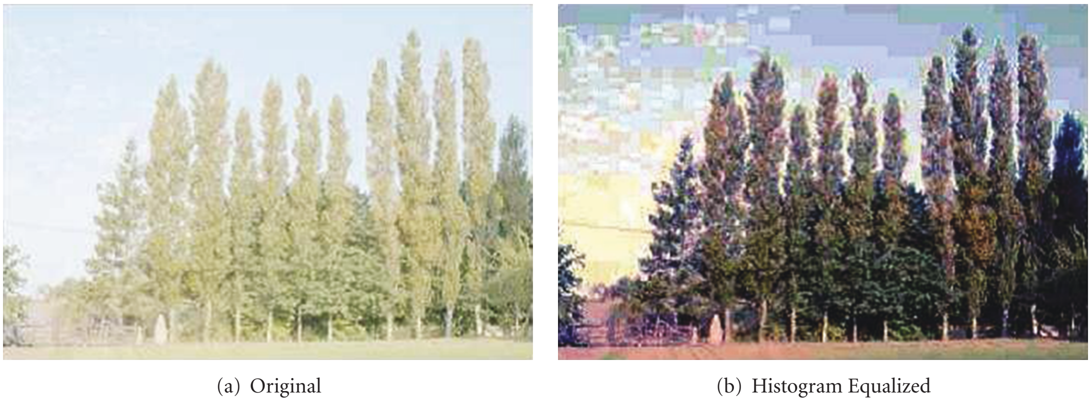
The human visual system (HVS) is a sophisticated mechanism capable of capturing a scene with very precise representation of detail and color. In the HVS, while individual sensing elements can only distinguish limited quantized levels, the entire system handles large dynamic range through various biological actions. Current capture or display devices cannot faithfully represent the entire dynamic range of the scene, therefore images taken from a camera or displayed on monitors/display devices suffer from certain limitations. As a result, bright regions of the image may appear overexposed and dark regions may appear underexposed. The objective of contrast enhancement is to improve the visual quality of images.

One of the most common techniques to enhance the contrast of images is to perform histogram equalization. The advantage of this technique is that it works very well for grayscale images; however, when histogram equalization is used to enhance color images, it may cause a shift in the color scale, resulting in artifacts and an imbalance in image color as shown in Figure 1. These unwanted artifacts are not desirable, as it is critical to maintain the color properties of the image while enhancing them.

A high-level overview of our approach can be seen in Figure 2. We assume any image to be a pixel-by-pixel product of the illumination (light that falls on the scene) and the reflectance component of the scene. This can be expressed as

$$\mathbf{I}(\mathbf{x}) = \mathbf{L}(\mathbf{x})\mathbf{R}(\mathbf{x}), \tag{1}$$

where $\mathbf{L}(\mathbf{x})$ is the illumination component and $\mathbf{R}(\mathbf{x})$ is the reflectance component of the image $\mathbf{I}(\mathbf{x})$ with spatial coordinate $\mathbf{x}$. In this paper, we deal with color images. So, $\mathbf{I}(\mathbf{x})$, $\mathbf{L}(\mathbf{x})$, and $\mathbf{R}(\mathbf{x})$ have 3 components—one for each color channel. For instance, for the illumination image, $\mathbf{L}(\mathbf{x})$, we denote the red color channel by $\overline{\mathbf{L}}_{\text{red}}(\mathbf{x})$, the green channel by $\overline{\mathbf{L}}_{\text{green}}(\mathbf{x})$, and the blue color channel by $\overline{\mathbf{L}}_{\text{blue}}(\mathbf{x})$. Similarly, we can denote the color channels for other images. The capital bold font denotes multiband. The presence of a bar above the capital bold font denotes singleband. Similarly, the spatial coordinate, $\mathbf{x} = (x, y) \in \Re^2$.


(a) Original

(b) Histogram Equalized

Figure 1: Histogram Equalization of color images. We can see artifacts in the sky and color shift along the edges of trees.

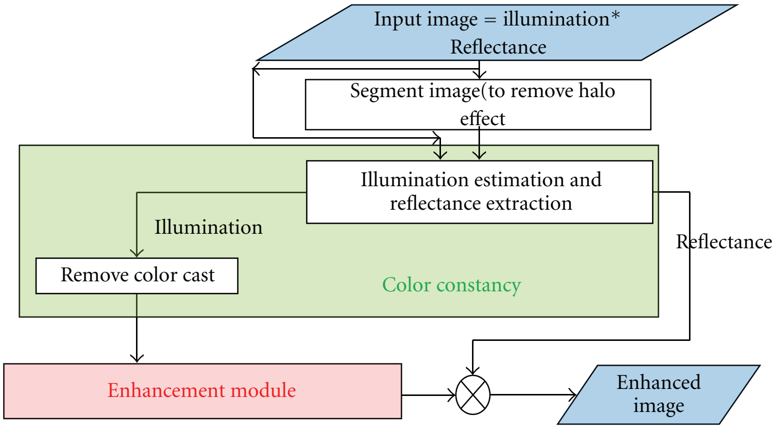


Figure 2: High-level overview of our method.

We first try to estimate the illumination component of the image, $\mathbf{L}(\mathbf{x})$, and thereby separate illumination from reflectance. The reason why we try to separate $\mathbf{L}(\mathbf{x})$ and $\mathbf{R}(\mathbf{x})$ is because experimentally Land [1] showed that the HVS can recognize the color in a scene irrespective of the illumination. This property of HVS is termed as *color constancy* and the theory is referred to as retinex theory. Our method also tries to mimic this human ability by removing the effects of the color of illumination from the image after estimating the illumination.

Trying to recover $\mathbf{L}(\mathbf{x})$ from an image and separate $\mathbf{L}(\mathbf{x})$ from $\mathbf{R}(\mathbf{x})$ is a mathematically ill-posed problem, so recovering $\mathbf{L}(\mathbf{x})$ needs further assumptions. We make the assumption that $\mathbf{L}(\mathbf{x})$ is smooth. In order to estimate $\mathbf{L}(\mathbf{x})$, based on this assumption, we smooth the image using denoising algorithms. Though this does not give us the true illumination of the scene, it gives us a good estimate of the illumination color. However, as a result of smoothing, strong halo effects can be seen along certain boundaries of the image. Therefore, before smoothing the image to estimate $\mathbf{L}(\mathbf{x})$, we preprocess the image and segment it. Smoothing can then be performed adaptively, and reduced around boundaries.

Once the color cast has been removed from the image (this is equivalent to achieving *White Balance*), we only process the $\mathbf{L}(\mathbf{x})$ component of the image. The motivation behind modifying only the illumination of the scene is to preserve the original properties of the scene—the reflectance of the scene. Also, the dynamic range of the illumination can be very large and hence we compress the dynamic range of the illumination. We modify the illumination of the image depending on the distribution of the intensity pixels, using logarithmic functions to estimate the enhanced illumination and then multiply it by the reflectance component to produce the enhanced image. This also helps in improving the local contrast of the image, which is another property of the human visual system. Our results show that even in complex nonuniform lighting conditions, the enhanced results look visually better.

The rest of the paper is organized as follows. In Section 2, we review previous work done in color constancy and in image enhancement. In Section 3, we describe our approach to achieve color constancy along with its evaluation on widely used large datasets of images. In Section 4, we describe the details of our algorithm with respect to separation of the illumination from reflectance and then processing of the

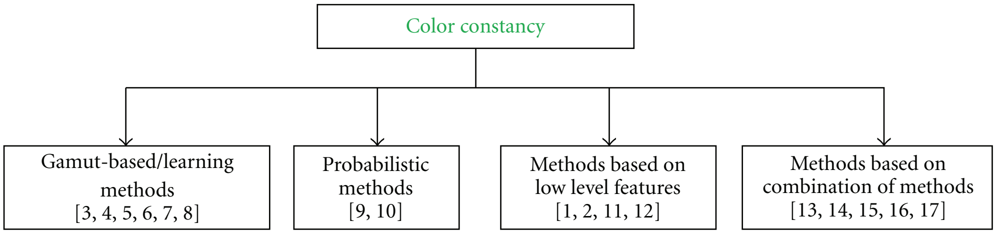

Color constancy
Gamut-based/learning methods [3, 4, 5, 6, 7, 8]
Probabilistic methods [9, 10]
Methods based on low level features [1, 2, 11, 12]
Methods based on combination of methods [13, 14, 15, 16, 17]

Figure 3: Summary of previous work in color constancy.

illumination component. In Section 5, we show the results of our proposed enhancement method and compare our results with other enhancement techniques including some state-of-the-art approaches, and results from commercial image processing packages. Finally, we conclude our paper in Section 6.

2. Previous Work

2.1. Color Constancy. Most color constancy algorithms make the assumption that only one light source is used to illuminate the scene. Given the image pixel values $\mathbf{I}(\mathbf{x})$, the goal of color constancy is to estimate the color of light source, assuming there is a single constant source for the entire image. Alternatively, as shown in [2], the illumination is considered to be a spectrum and assuming that the color of illuminant $\mathbf{l}$ depends on the illuminant spectral power distribution $L(\lambda)$ and camera sensitivity $\mathbf{c}(\lambda)$, $\mathbf{l}$ is estimated as

$$\mathbf{l} = \int_{\omega} L(\lambda)\mathbf{c}(\lambda)d\lambda, \tag{2}$$

where $\mathbf{l}$ is the estimated color of the illuminant, ω is the visible spectrum and λ is the wavelength. Therefore, estimating $\mathbf{l}$ is equivalent to estimating its color coordinates, where $\mathbf{l}$ has 3 components—one value for each color channel.

Many color constancy algorithms have been proposed to estimate the color of illumination of a scene. These methods can be broadly classified into 4 categories: (1) gamut-based/learning methods, (2) probabilistic methods, (3) methods based on low-level features, and (4) methods based on combination of different methods. The literature of these methods can be summarized as shown in Figure 3.

The gamut mapping method proposed by Forsyth [3] is based on the observation that given an illuminant, the range of RGB values present in a scene is limited. Under a known illuminant (typically, white), the set of all RGB colors is inscribed inside a convex hull and is called the *canonical* gamut. This method tries to estimate the illuminant color by finding an appropriate mapping from an image gamut to the *canonical* gamut. Since this method could result in infeasible solutions, Finlayson et al. [4] improve the above algorithm by constraining the transformations so that the illuminant estimation corresponds to a predefined set of illuminants. Finlayson et al. [5] use the knowledge about appearance of colors under a certain illumination as a prior to estimate the probability of an illuminant from a set of illuminations. The disadvantage of this method is that the estimation of the illuminant depends on a good model of the lights and surfaces, which is not easily available. Chakrabarti et al. [6] consider the spatial dependencies between pixels to estimate the illuminant color. Another learning-based approach by Cardei et al. [7] use a neural network to learn the illumination of a scene from a large number of training data. The disadvantage with neural networks is that the choice of training dataset heavily influences the estimation of the illuminant color. A nonparametric linear regression tool called kernel regression has also been used to estimate the illuminant chromaticity [8].

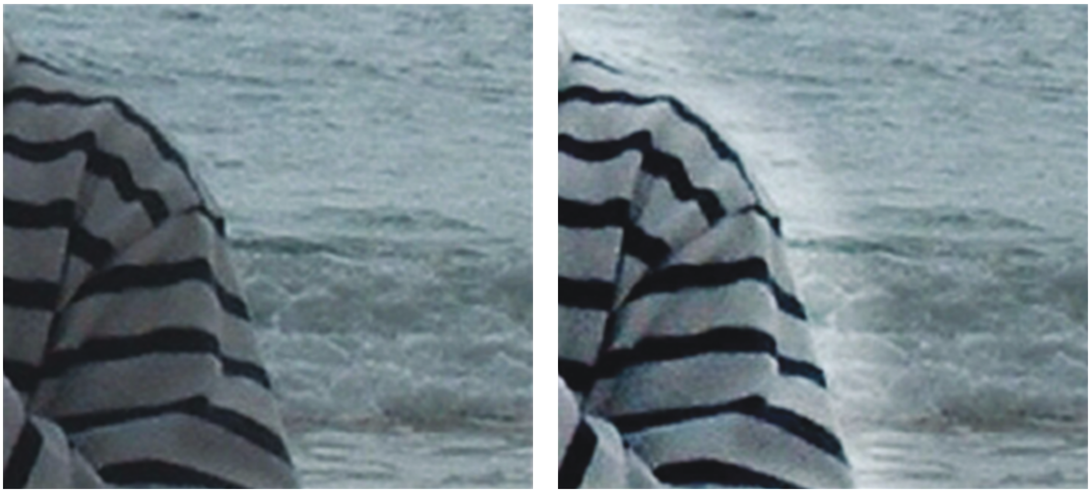
Probabilistic methods include Bayesian approaches [9] that estimate the illuminant color depending on the posterior distribution of the image data. These methods first model the relation between the illuminants and surfaces in a scene. They create a prior distribution depending on the probability of the existence of a particular illuminant or surface in a scene, and then using Bayes's rule, compute the posterior distribution. Rosenberg et al. [10] combine the information about neighboring pixels being correlated [5] within the Bayesian framework.

A disadvantage of the above-mentioned algorithms is that they are quite complex and all the methods require large datasets of images with known values of illumination for training or as a prior. Also, the performance may be influenced by the choice of training dataset.

Another set of methods uses low level features of the image. The white-patch assumption [1] is a simple method that estimates the illuminant value by measuring the maximum value in each of the color channels. The grey-world algorithm [11] assumes that the average pixel value of a scene is grey. The grey-edge algorithm [2] measures the derivative of the image and assumes the average edge difference to be achromatic. As shown in [2], all the above low level color constancy methods can be expressed as

$$\left(\int \left|\frac{\partial^n \mathbf{I}^\sigma(\mathbf{x})}{\partial \mathbf{x}^n}\right|^p d\mathbf{x}\right)^{1/p} = k\mathbf{l}^{n,p,\sigma}, \tag{3}$$

where n is the order of derivative, p is the Minkowski norm and σ is the parameter for smoothing the image $\mathbf{I}(\mathbf{x})$ with a Gaussian filter. The original formulation of (3) for the white-patch assumption and the grey-world algorithm can be found in [12]. As expressed in (3), the white-patch assumption can be expressed as $\mathbf{l}^{0,\infty,0}$, the grey-world algorithm can be expressed as $\mathbf{l}^{0,1,0}$ and the n-order grey-edge algorithm can be expressed as $\mathbf{l}^{n,p,\sigma}$. Van de Weijer et al. [2] have shown results using values of $n = 1$ and $n = 2$.



(a) Original (b) Enhanced

Figure 4: Halo effect caused during retinex enhancement can be observed around the edge of the body and the background.

More recent techniques to achieve color constancy use higher-level information and also use combinations of different existing color constancy methods. Reference [13] estimates the illuminant by taking a weighted average of different methods. The weights are predetermined depending on the choice of dataset. Gijsenij and Gevers [14] use Weibull parameterization to get the characteristics of the image and depending on those values, divide the image space into clusters using k-means algorithm and then use the best color constancy algorithm corresponding to that cluster. The best algorithm for a cluster is learnt from the training dataset. Van de Weijer et al. [15] model an image as a combination of various semantic classes such as sky, grass, road and buildings. Depending on the likelihood of semantic content, the illuminant color is estimated corresponding to the different classes. Similarly, information about images being indoor or outdoor are also used to select a color constancy algorithm and consequently estimate the color of illuminant [16]. More recently, 3D scene geometry is used to classify images and a color constancy algorithm is chosen according to the classification results to estimate the illuminant color [17].

2.2. Enhancement Techniques. The most common technique to enhance images is to equalize the global histogram or to perform a global stretch [18]. Since this does not always gives us good results, local approaches to histogram equalization have been proposed. One such technique uses adaptive histogram equalization [18] that computes the intensity value for a pixel based on the local histogram for a local window. Another technique to obtain image enhancement is by using curvelet transformations [19]. A recent technique has been proposed by Palma-Amestoy et al. [20] that uses a variational framework based on human perception for achieving enhancement. This method also removes the color cast from the images during image enhancement unlike most other methods and achieves the same goal as our algorithm. Inspired by the grey-world algorithm [11], Rizzi et al. [21] introduced a technique called automatic color equalization (ACE) that uses an odd function of differences between pixel intensities to enhance an image. This is a two step process—the first step computes the chromatic spatial adjustment by considering the difference in the pixels and weighted by a distance function. The second step maximizes the dynamic range of the image. A technique inspired from the retinex theory [1] is the random spray retinex (RSR) by Provenzi et al. [22] that uses local information within the retinex theory framework by replacing paths with a random 2-D pixel spray around a given pixel under consideration. The previous two techniques were fused by Provenzi et al. [23] and called RACE that account for the defects of those two algorithms (RSR has good saturation properties but cannot recover detail in dark regions whereas ACE has good detail recovery in dark regions but tends to wash out images). Another recent technique based on the retinex theory is the Kernel-based retinex (KBR) [24] that is based on computing a kernel function that represents the probability density of picking a pixel $\mathbf{y}$ in the neighborhood of another pixel $\mathbf{x}$ where $\mathbf{x}$ is fixed and $\mathbf{y}$ could be any pixel in the image.

Other methods that can enhance images under difficult lighting conditions have been inspired from the retinex theory. One such popular technique was proposed by Jobson et al. [25] called multi scale retinex with color restoration (MSRCR) where the color value of a pixel is computated by taking the ratio of the pixel to the weighted average of the neighboring pixels. One disadvantage of this technique is that there could be abnormal color shifts because three color channels are treated independently. An inherent problem in most retinex implementation is the strong halo effect in regions having strong contrast. The halo effects are shown in Figure 4. The halo effects are reduced in another variation of the retinex theory that is proposed by Kimmel et al. [26] where the illumination component is removed from the reflectance component of the image using a variational approach. While enhancing the illumination channel, the technique uses only gamma-correction and the method is not automated. The user has to manually modify the value of gamma depending on the exposure conditions of the image—the value of gamma for enhancing underexposed images is very different from the value of gamma for enhancing overexposed images. A disadvantage of the techniques that are purely based on the retinex theory is that these techniques cannot enhance overexposed images because of the inherent nature of the retinex theory to always increase the pixel intensities as shown in [27]. However, this modification is tackled in both [20, 24].

A recent method to obtain automated image enhancement is the technique proposed by Tao et al. [28] that uses an inverse sigmoid function. Due to lack of flexibility of the inverse sigmoid curve, this technique does not always result in achieving good dynamic range compression. There are other techniques that have been designed in the field of computer graphics for high dynamic range images to obtain improved color contrast enhancement. Pattanaik et al. [29] use a tone-mapping function that tries to mimic the processing of the human visual system. It applies a local filtering procedure and uses a multiscale mechanism. This technique is a local approach and it may result in halo effects. Another technique proposed by Larson et al. [30] does tone-mapping based on iterative histogram modification. Ashikhmin [31] has also proposed a tone-mapping algorithm for enhancement and is based on a Gaussian pyramid framework.

3. Our Approach to Achieve Color Constancy

As stated earlier, color constancy is the human ability to perceive the color of a scene irrespective of the illumination conditions. Therefore, to achieve color constancy, we should estimate the color of illumination. In order to estimate the color of the illumination of an image, we first process the image to estimate the illumination component, $\mathbf{L}(\mathbf{x})$. We make the assumption that $\mathbf{L}(\mathbf{x})$ is smooth. Based on this assumption, we use denoising techniques to process the image and smooth it. Denoising techniques are traditionally used in image processing to remove noise from the image. This involves smoothing of the images and hence removal of the noise. Because of our smoothness assumption of $\mathbf{L}(\mathbf{x})$, the smooth image that is obtained by denoising, is our estimate of the illumination image, $\mathbf{L}(\mathbf{x})$. Once we have estimated $\mathbf{L}(\mathbf{x})$, we apply the white-patch Assumption (maximum in each color channel), as described in Section 3.2 on $\mathbf{L}(\mathbf{x})$ to estimate the color of the illumination.

In the most simple filtering example, the smoothing operation calculates the average of a region around every pixel of the image. As the size of the region increases (limited by the size of the image), the estimate of the maximum value in each color channel tends towards the average value in each color channel. Therefore, in a way that Minkowski norm works as mentioned in Section 2, the smoothing operation unifies the White Patch Assumption [1] and the grey-world algorithm [11] in one approach. Hence, similar to (3), we also assume the true color of illumination to be somewhere between the maximum and the mean of the color channels. These trends in the estimate of the illumination color while filtering using different parameters are explained in Section 3.3.2.

3.1. Denoising Techniques. We study 4 different existing denoising techniques to smooth the image: (1) Gaussian filter, (2) median filter, (3) bilateral filter, and (4) nonLocal means filter. These filters have different levels of complexity and vary in their smoothness mechanism. We consider the input image $\mathbf{I}(\mathbf{x})$ to be defined on a bounded domain $\Omega \subset \Re^2$ and let $\mathbf{x} \in \Omega$. The description of these filters is given below.

3.1.1. Gaussian Filter. Blurring the image removes noise. This filter can be expressed as

$$\mathbf{G}(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{(-|\mathbf{x}|^2/2\sigma^2)}, \quad (4)$$

where σ is the smoothing parameter. This is a 2D Gaussian radially symmetric kernel where $|\mathbf{x}| = (x^2 + y^2)^{1/2}$.

Gijsenij and Gevers [32] have explored iterated local averaging and can be considered similar to the Gaussian filter approach. The edge information is lost during Gaussian filter and this introduces error while estimating the illuminant. We apply this filter across all 3 color channels (red, green and blue) of the image.

3.1.2. Median Filter. For every pixel in an image, the median filter [33] chooses the median color value amongst the neighborhood pixels in a window $\mathbf{W}$ for that pixel—every pixel has same number of color values above and below it. Smoothing using median filter may result in the loss of fine details of the image though boundaries may be preserved. We apply this filter on all 3 color channels (red, green and blue) of the image.

3.1.3. Bilateral Filter. In case of bilateral filter [34], every pixel of the image is replaced by a weighted sum of its neighbors. The neighboring pixels are chosen from a window $(\mathbf{W})$ around a given pixel. The weights depend on two parameters: (1) Proximity of the neighboring pixels to the current pixel (Closeness function) and (2) similarity of the neighboring pixels to the current pixel (similarity function). The closer and the more similar pixels are given higher weights. The two parameters can be combined to describe the bilateral filter as follows:

$$\mathbf{L}(\mathbf{x}) = k^{-1}(\mathbf{x}) \int_{\mathbf{W}(\mathbf{x})} c(\mathbf{y}, \mathbf{x}) s(\mathbf{I}(\mathbf{y}), \mathbf{I}(\mathbf{x})) \mathbf{I}(\mathbf{y}) d\mathbf{y}, \quad (5)$$

where $\mathbf{y}$ is the neighboring pixel and the normalization term k can be given by:

$$k(\mathbf{x}) = \int_{\mathbf{W}(\mathbf{x})} c(\mathbf{y}, \mathbf{x}) s(\mathbf{I}(\mathbf{y}), \mathbf{I}(\mathbf{x})) d\mathbf{y}. \quad (6)$$

The closeness function is

$$c(\mathbf{y}, \mathbf{x}) = e^{-(1/2)(d(\mathbf{y},\mathbf{x})/\sigma_c)^2}, \quad (7)$$

where $d(\mathbf{y}, \mathbf{x}) = \|\mathbf{y} - \mathbf{x}\|$ is the Euclidean distance between a given pixel, $\mathbf{x}$ and its neighbor $\mathbf{y}$. The similarity function is

$$s(\mathbf{I}(\mathbf{y}), \mathbf{I}(\mathbf{x})) = e^{-(1/2)(\delta(\mathbf{I}(\mathbf{y}), \mathbf{I}(\mathbf{x}))/\sigma_s)^2}, \tag{8}$$

where $\delta(\mathbf{I}(\mathbf{y}), \mathbf{I}(\mathbf{x}))$ is the pixel value difference between $\mathbf{x}$ and $\mathbf{y}$. The closeness function from (7) having standard deviation σ_c and the similarity function from (8) having standard deviation σ_s are Gaussian functions of the Euclidean distance between their parameters. In the original implementation of the bilateral filter, Tomasi and Manduchi [34] have used the CIELAB color space. We present results by applying this filter on all 3 channels of the CIELAB color space of the image. Applying this filter on the RGB colorspace instead does not result in a significant difference.

3.1.4. Nonlocal Means Filter. The hypothesis behind nonlocal means (NL-means) technique [35] is that for any image, the most similar pixels to a given pixel need not be close to it. They could lie anywhere in the image. For comparing how similar the pixels are, instead of checking the difference between the pixel values (which is used in bilateral filtering), the neighborhood of the pixel is considered—that is, comparison of a window around the pixel is done. This technique uses self-similarity in an image to reduce the noise. The formulation of the NL-means filter is:

$$\begin{aligned} \mathrm{NL}(\mathbf{I}(\mathbf{x})) &= \mathbf{L}(\mathbf{x}) \\ &= \frac{1}{N(\mathbf{x})} \int_{\Omega} e^{-(G_\rho * |\mathbf{I}(\mathbf{x}+\cdot) - \mathbf{I}(\mathbf{y}+\cdot)|^2)(0)/h^2} \mathbf{I}(\mathbf{y})\, d\mathbf{y}, \end{aligned} \tag{9}$$

where $\mathbf{y} \in \Omega$, $\mathbf{I}(\mathbf{x})$ is the observed intensity at pixel $\mathbf{x}$, G_ρ is a Gaussian kernel with standard deviation ρ, h is the filtering parameter that controls the amount of smoothing and $N(\mathbf{x})$ is the normalizing factor. Equation (9) means that an image pixel value $\mathbf{I}(\mathbf{x})$ is replaced by the weighted average of other pixel values in the image $\mathbf{I}(\mathbf{y})$. The weights are significant only if a window around pixel $\mathbf{x}$ looks like the corresponding window around pixel $\mathbf{y}$. While comparing the windows, we consider the Euclidean distance between the 2 windows. However, we weigh this distance by a Gaussian-like kernel decaying from the center of the window to its boundaries. This is because closer pixels are more related, and so pixels closer to the reference pixel are given more importance. Ideally, we should search the entire image to find a similar neighborhood. But for efficient computation, we consider a smaller local search area, $\mathbf{S}$. The numerator of the exponential accounts for the neighborhood of a pixel which we denote by $\mathbf{W}$. Please refer to [35] for a more detailed explanation of the NL-means filter. For discrete images, the integral over Ω can be replaced by a summation over all pixels of the image. We apply this filter on all 3 color channels (red, green and blue) of the image.

3.2. Illuminant Color Estimation. Once we have obtained the illumination image $\mathbf{L}(\mathbf{x})$, by smoothing the original color image $\mathbf{I}(\mathbf{x})$, in order to estimate the color of the illumination $\mathbf{l}$, we use the white-patch assumption [1]. This method computes the maximum values of each of the 3 color channels—red, green and blue. The proposed formulation for $\mathbf{l}$ is

$$\mathbf{l} = \left[\max_{\mathbf{x}} (\overline{\mathbf{L}}_{\text{red}}(\mathbf{x})), \max_{\mathbf{x}} \left(\overline{\mathbf{L}}_{\text{green}}(\mathbf{x})\right), \max_{\mathbf{x}} (\overline{\mathbf{L}}_{\text{blue}}(\mathbf{x})) \right], \tag{10}$$

where $\overline{\mathbf{L}}_{\text{red}}(\mathbf{x})$, $\overline{\mathbf{L}}_{\text{green}}(\mathbf{x})$ and $\overline{\mathbf{L}}_{\text{blue}}(\mathbf{x})$ are the red, green, and blue color channels of the denoised image (illumination image) and the max operation is performed on the separate color channels. The maximum value of these color channels need not lie at the same pixel of the image. This estimation is based on the hypothesis that since a white patch of the image reflects all the light that is incident on it, its position can be found by searching for the maximum values of the red, green and blue channels. The $\mathbf{l}$ vector thus estimated is normalized and is denoted by $\hat{\mathbf{l}}$, which has 3 components: $\hat{\mathbf{l}}(1)$, $\hat{\mathbf{l}}(2)$ and $\hat{\mathbf{l}}(3)$ for red, green, and blue respectively.

Once the color of the illumination is estimated as described above, we remove the effect of color cast from the image. This is equivalent to adding white light to the scene. This can be represented as

$$\begin{aligned} \overline{\mathbf{L}}_{\text{red}_{\text{color-corrected}}}(\mathbf{x}) &= \frac{1}{\sqrt{3}} \frac{\overline{\mathbf{L}}_{\text{red}}(\mathbf{x})}{\hat{\mathbf{l}}(1)}, \\ \overline{\mathbf{L}}_{\text{green}_{\text{color-corrected}}}(\mathbf{x}) &= \frac{1}{\sqrt{3}} \frac{\overline{\mathbf{L}}_{\text{green}}(\mathbf{x})}{\hat{\mathbf{l}}(2)}, \\ \overline{\mathbf{L}}_{\text{blue}_{\text{color-corrected}}}(\mathbf{x}) &= \frac{1}{\sqrt{3}} \frac{\overline{\mathbf{L}}_{\text{blue}}(\mathbf{x})}{\hat{\mathbf{l}}(3)}, \end{aligned} \tag{11}$$

where the factor $\sqrt{3}$ is the normalization constant based on the diagonal model to preserve the intensity of pixels. Combining all 3 color channels—$\overline{\mathbf{L}}_{\text{red}_{\text{color-corrected}}}(\mathbf{x})$, $\overline{\mathbf{L}}_{\text{green}_{\text{color-corrected}}}(\mathbf{x})$ and $\overline{\mathbf{L}}_{\text{blue}_{\text{color-corrected}}}(\mathbf{x})$, shown in (11)—gives us the color corrected image. This is how we obtain the images illustrated in Figure 12.

3.3. Evaluation and Discussion. In order to evaluate our approach, we conduct experiments on two widely used datasets. The ground-truth values of the illuminant of the scenes are provided for both datasets. For quantitative evaluation of the approaches, the error difference between the estimated illumination color $\mathbf{l}$ and the ground truth illumination color $\mathbf{l}_{\text{gt}}$ is computed. We use the angular error as a measure and it can be computed as

$$\text{angular error, } \epsilon = \cos^{-1}\left(\hat{\mathbf{l}} \cdot \hat{\mathbf{l}}_{\text{gt}}\right), \tag{12}$$

where $(\hat{\cdot})$ stands for normalized values. In order to measure the overall performance across each of the datasets, the median of the angular errors is considered as a suitable measure [36].

3.3.1. Controlled Indoor Environment Dataset. This dataset [37] consists of a controlled indoor environment with 30 different scenes taken under 11 different illuminant conditions. All the images are illuminated by only one illuminant. Some of these images were deemed unusable

(a) (b) (c)

Figure 5: Images from the controlled indoor environment [37].

Table 1: Error for the controlled indoor environment using various color constancy methods. The parameters of our approach have been described in Section 3.1. The best result is reported in bold.

Method	Parameters	Median ϵ ($^\circ$)	RMS_{rg}
White-patch	—	6.4	0.053
Grey-world	—	6.9	—
1st-order grey edge	$p = 7$	3.2	—
2nd-order grey edge	$p = 7$	2.8	—
Color by correlation	—	3.1	0.061
Gamut mapping	—	2.9	—
Neural networks	—	7.7	0.071
GCIE version 3, 11 lights	—	1.3	—
GCIE version 3, 87 lights	—	2.6	—
Kernel regression	—	—	0.052
Support vector machines	—	—	0.066
Gaussian filter	$\sigma = 3, x = 30$	2.9	0.043
Median filter	$\mathbf{W} = 20$	**2.4**	0.044
Bilateral filter	$\sigma_c = 2, \sigma_s = 5, \mathbf{W} = 5$	2.8	0.044
NL-means filter	$h = 0.2, \mathbf{S} = 3, \mathbf{W} = 2$	2.5	0.043

by the original authors and therefore, this dataset has 321 images. Some sample images from this dataset can be seen in Figure 5. All images have resolution of 637 × 468 pixels.

The results of existing color constancy algorithms on this dataset are summarized in Table 1. These results are available in [2, 4, 36]. Some methods in the literature [8] have used the root mean square (RMS) error between the estimated illuminantion chromaticity, $\mathbf{lc}_e$ and the actual illumination chromaticity, $\mathbf{lc}_{\text{gt}}$ to evaluate their results although this is not the best metric [36]. Since we do not have access to the individual error values, we compare the results as is. The RMS_{rg} error can be calculated as

$$\text{RMS}_{\text{rg}} = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}\left(\mathbf{lc}_i(j) - \mathbf{lc}_{\text{gt}_i}(j)\right)^2\right)^{1/2}, \qquad (13)$$

where N is the total number of images, i is the index of images and M is the number of color channels ($M = 2$ for chromaticity space). We calculate error for the rg space. The chromaticity for r and g for the estimated illuminant can be computed as $\mathbf{lc}(1) = \hat{\mathbf{I}}(1)/(\hat{\mathbf{I}}(1) + \hat{\mathbf{I}}(2) + \hat{\mathbf{I}}(3))$ and $\mathbf{lc}(2) = \hat{\mathbf{I}}(2)/(\hat{\mathbf{I}}(1) + \hat{\mathbf{I}}(2) + \hat{\mathbf{I}}(3))$. Similarly, we can compute the chromaticity for the ground truth illuminant. The RMS_{rg} error for white-patch, Neural Networks and Color by Correlation were presented in [38]. The performance of our approach on this dataset are also presented in Table 1.

From Table 1, we can see that our approach gives the best performance both with respect to the angular error and the RMS_{rg} error. On comparing the error of our approach with that of all the approaches in [38], we can see that our approach has the least RMS_{rg} error. The GCIE algorithm using 11 lights perform very well because this technique uses the 11 illuminants as a prior knowledge. It constrains the estimated value of the illuminant to lie in that set. However, the performace drops if more illuminants are used for training. We also compared our results with a recent denoising technique by Dabov et al. [39]. However, their technique produces a median error of 5.67° on this dataset which is worse than our results.

3.3.2. Effects of Parameter Modification. In this section, we discuss the effects of parameter setting for the different methods of our approach. We use the images from controlled indoor environment dataset and plot the median angular error across the dataset against the parameters of the filter.

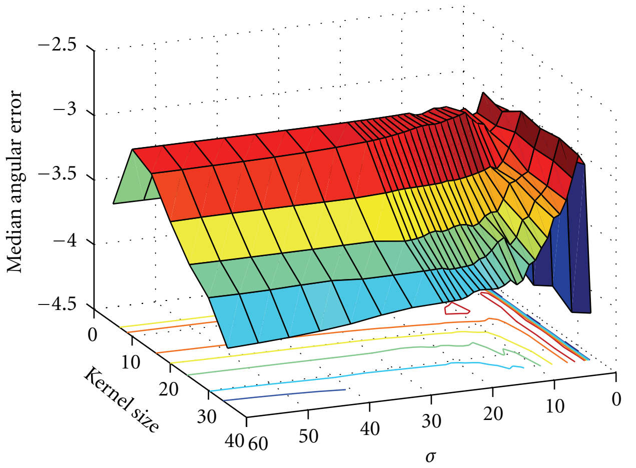


Figure 6: Median angular error as a function of σ and kernel size for a Gaussian filter.

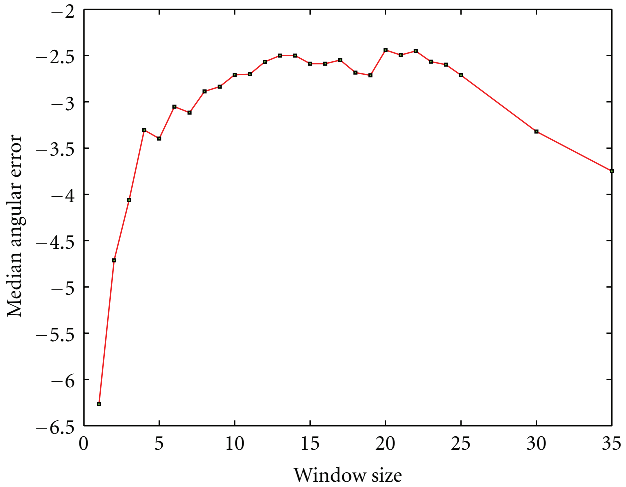


Figure 7: Median angular error as a function of window size for a median filter.

The axis for the median angular error is inverted for better visualization.

The error plot for Gaussian filter is shown in Figure 6. Smoothing with $\sigma = 0$ will have no effect and therefore, for $\sigma = 0$, this method works like the white-patch assumption. However, as $\sigma \rightarrow \infty$, the algorithm is equivalent to taking the average of the image and therefore, the algorithm is equivalent to the grey-world algorithm. We obtain the best results for an intermediate value of σ. Also from the plot, we can see that for lower values of σ, the median angular error remains fairly consistent for higher values of window size. This is because in case of a Gaussian distribution, 99.7% pixels values lie within 3σ of the mean and therefore higher values of window size does not affect the calculation of a pixel value. Also, for higher values of σ, we can see that the error value increases with increase in window size. Irrespective of the σ values, typically window size of around 5–20 pixels gives us best results.

The error plot for median filter is shown in Figure 7. As can be seen in the case of Gaussian filter, we can see that we obtain best error values for window size (**W**) of around 5–20 pixels. For $\mathbf{W} = 0$, this algorithm is once again the same as white-patch algorithm. However when the value of $\mathbf{W} \rightarrow \infty$, the color estimated by this method will be the median color value of all the pixels present in the image. Realistically, the limiting factor for **W** will be the image size. For this dataset, the median angular error when **W** is the image size, is **12.99°**.

The error plot for Nonlocal means filter is shown in Figure 8. In case of a Nonlocal means filter, the amount of smoothing of the image depends on the filtering parameter, h. A higher value of h does more smoothing of the image. As we can see in Figure 8, the error value converges for higher values of h. For $h = 0$, the algorithm will not perform any smoothing and this method will be the same as the white-patch algorithm.

The error plots for bilateral filter are shown in Figures 9 and 10. The left image of Figure 9 plots the median angular error for low values of σ for spatial domain, σ_c and σ for intensity domain, σ_s whereas the right image plots the median angular error for high values of σ_c and σ_s. As with the Nonlocal means filter, we can see that the error value converges for high smoothing (high values of σ_c and σ_s). For values of σ's = 0, this algorithm will behave like the White Patch algorithm. The best results are obtained from an intermediate value of smoothing parameter.

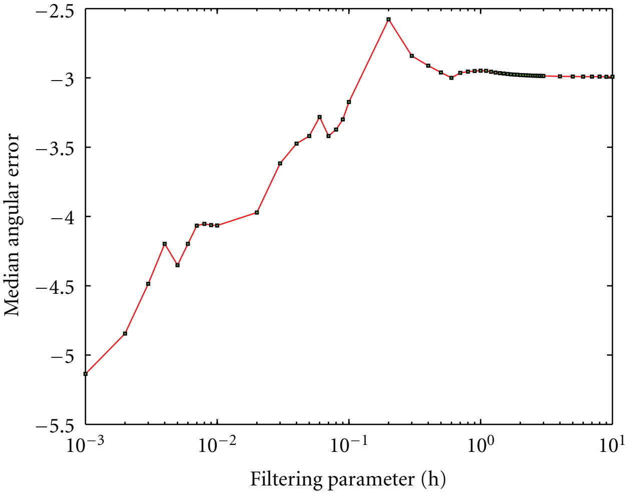


Figure 8: Median angular error as a function of filtering parameter for a Nonlocal means filter.

As shown in Figure 10, given the best values of σ_c and σ_s, we see the effects of changing the window size on the median errors. For low window size, since there is not enough information, the error values are high. However, since the individual closeness and similarity functions are Gaussian distributions, having a very high window size does not help much and the error value converges close to the best value.

3.3.3. Real-World Environment Dataset. This dataset [40] consists of approximately 11000 images from 15 different scenes taken in a real-world environment. Some sample

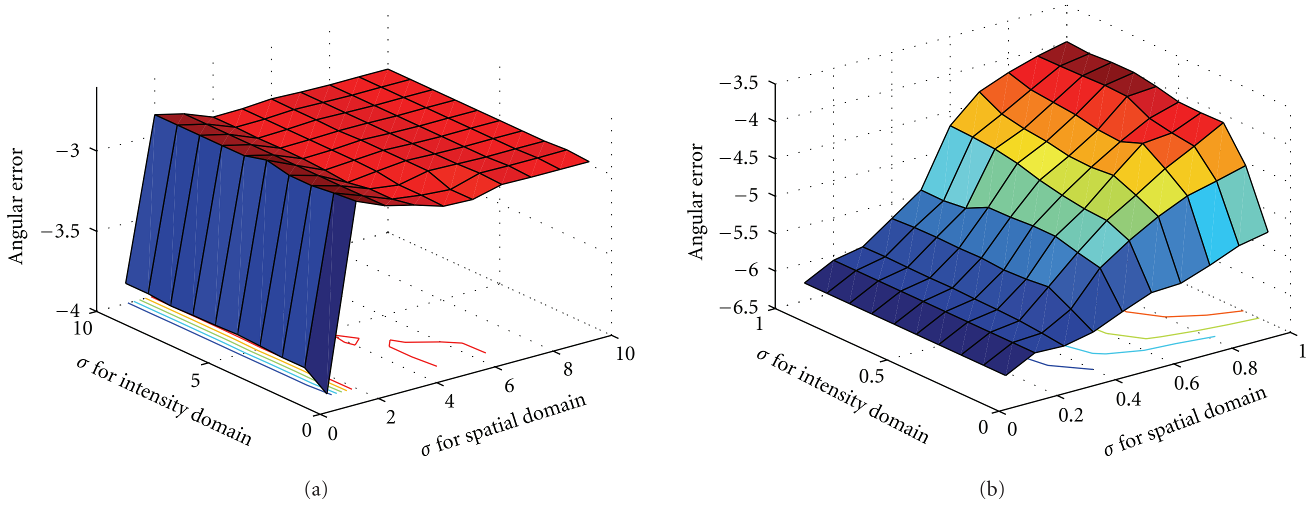


Figure 9: Median angular error as a function of σ's for a bilateral filter.

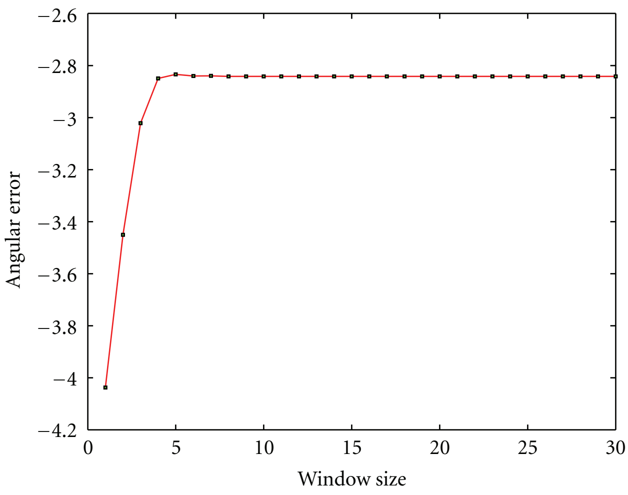


Figure 10: Median angular error as a function of window size for a bilateral filter.

images from this dataset can be seen in Figure 11. The images include both indoor and outdoor scenes. All images have 360 × 240 resolution as they are all extracted from a digital video. As a result, there is a high correlation between the images from one scene in the database. Therefore, from each scene we randomly select 10 images amounting to 150 images in total on which we test our approach. Some sample images from this dataset can be seen in Figure 11.

As can be seen in Figure 11, all the images from this dataset have a grey ball at the bottom right corner of the image. This grey ball was mounted on the video camera to estimate the ground truth value of the color of the illuminant and those values are available with the dataset. However, while trying to estimate the illuminant color, we exclude the bottom right quadrant as depicted by the white box in Figure 12.

The results of existing color constancy algorithms and our approaches on this dataset can be found in Table 2. The selection of parameters significantly improves the results when compared with the state-of-the-art color constancy approaches. From Table 2, we can see that our method have a **23**% improvement over the state-of-the-art grey-edge algorithm. We also compared our algorithm with a very recent technique—"Beyond Bag of pixels" approach [6] and found that our method gives us almost **26**% improvement. On comparing our results with a recent denoising technique by Dabov et al. [39], that produces a median error of 3.96°, we obtain an improvement of almost **14**%.

3.3.4. Discussion. From the experiments that we have conducted on the 2 widely used datasets and the results presented in Tables 1 and 2, we can see that our approach gives us results that are better than state-of-the-art color constancy approaches. On the controlled indoor environment dataset, [2] have reported a median error of 3.2° for the 1st order grey-edge algorithm with $p = 7$ and preprocessing with a Gaussian filter of $\sigma = 4$. The median error for the same dataset for the 2nd order grey-edge algorithm with $p = 7$ and preprocessing with a Gaussian filter of $\sigma = 5$ is 2.7°. Our experiments have shown that using just a Gaussian filter with $\sigma = 4$ gives us a median error of 3.08° and with $\sigma = 5$ gives us a median error of 3.12°. This makes us wonder if applying 1st order grey-edge algorithm hurts the illuminant color estimation. However there are benefits of using higher order grey-edge color constancy algorithms. It will be interesting to explore how grey-edge color constancy algorithms with order $n > 2$ affect the estimation of illuminant color. It will also be interesting to observe how complex methods such as GCIE, neural networks and other learning methods perform on images that are already preprocessed by our approach.

For the denoising algorithms, we observe that the window size required for the controlled indoor image dataset is larger than the one required for the real-world environment

(a) (b) (c)

FIGURE 11: Images from the real-world environment [40].

TABLE 2: Median angular error for the real-world environment. The parameters of our approach have been described in Section 3.1. The best result is reported in bold.

Method	Parameters	Median ϵ (°)
White-patch	—	4.85
Grey world	—	7.36
1st-order grey edge	$p = 6$	4.41
Beyond bags of pixels	—	4.58
Gaussian filter	$\sigma = 5, x = 15$	3.63
Median filter	$\mathbf{W} = 9$	3.42
Bilateral filter	$\sigma_c = 5, \sigma_s = 7, \mathbf{W} = 3$	3.4
NL-means filter	$h = 1.0, \mathbf{S} = 3, \mathbf{W} = 2$	**3.39**

TABLE 3: Ranks of algorithm according to Wilcoxon signed rank Test. The best results are reported in bold.

Method	WSTs
White-patch	1
Grey world	0
1st-order grey edge	2
Gaussian filter	2
Median filter	2
Bilateral filter	**3**
NL-means filter	**3**

dataset. This could be because the controlled environment dataset does not have too much variability—it consists of an object or two with/without a background. However, for the real-world dataset, there is a lot of variability and so more information is available even in a small window.

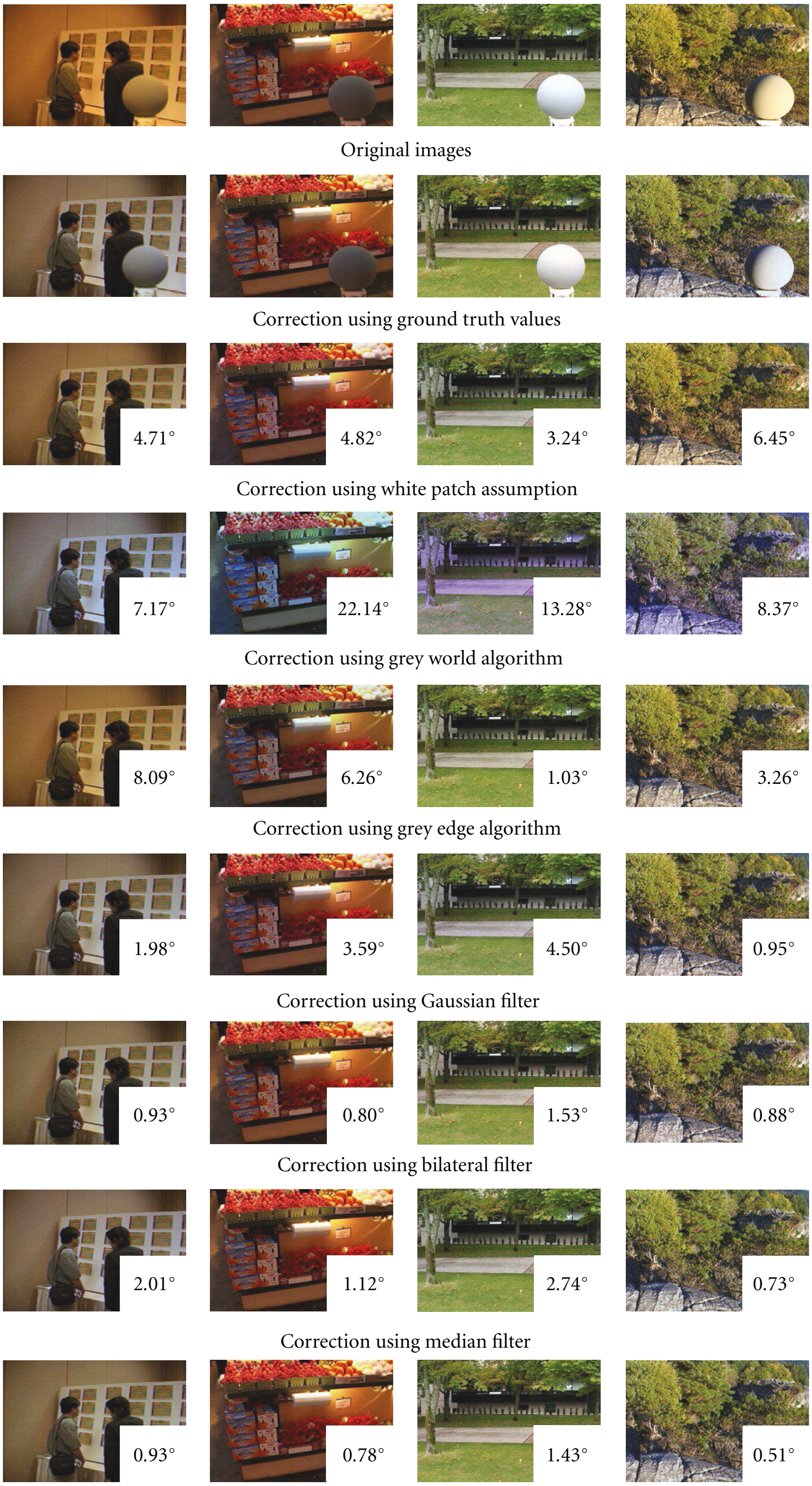
As shown in [35], amongst the filters described here, NL-means filter performs the best denoising. It best preserves the structure of the image while denoising the image. We observe a similar correspondence between the denoising capabilities of the filter and its illumination color estimation. The best illumination estimation results are obtained by using the best denoising filters.

In order to compare the performance of different color constancy algorithms, we use the Wilcoxon signed-rank test [41]. For a given dataset (We choose the real-world environment dataset), let C_1 and C_2 denote the angular error between the illuminant estimation of two different algorithms and the ground truth illuminant values. Let the medians of these 2 angular errors be m_{c_1} and m_{c_2}. The Wilcoxon signed-rank test is used to test the null hypothesis $H_0 : m_{c_1} = m_{c_2}$. In order to test this hypothesis, for each of N images, the angular error difference is considered—$(e_{c_1}^1 - e_{c_2}^1), (e_{c_1}^2 - e_{c_2}^2), \ldots, (e_{c_1}^N - e_{c_2}^N)$. These error pairs are ranked according to their absolute differences. If the hypothesis H_0 is correct, the sum of the ranks is 0. However, if the sum of ranks is different from 0, we consider the alternate hypothesis $H_1 : m_{c_1} < m_{c_2}$ to be true. We reject/accept the hypothesis if the probability of observing the error differences is less than or equal to a given significance level α. We compare every color constancy algorithm with every other color constancy algorithm and generate a score that tells us the number of times a given algorithm has been considered to be *significantly* better than the others. The results are presented in Table 3.

4. Our Image Enhancement Method

Our method for image enhancement is motivated by illuminance-reflectance modeling. It consists of 2 key steps: (1) Illumination estimation using Nonlocal means technique (2) Automatic enhancement of illumination. The flowchart of our enhancement module is shown in Figure 13. For our illumination estimation, we use the Nonlocal means(NL-means) technique because as shown in Section 3.3.3, it performs amongst the best while trying to estimate the color of illumination. Also as shown in [35], NL-means filter does a very good job at preserving the original structure of the image while smoothing the image.

4.1. Estimating Illumination Using Nonlocal Means Technique. As shown in Section 3.1.4, we smooth the image using Nonlocal means filter to estimate the illumination component of the image, $\mathbf{L}(\mathbf{x})$. However, while smoothing an image during illumination separation, a potential artifact that may occur is the halo artifact. Existing enhancement techniques that use illumination separation suffer from this drawback. This happens across the edge of regions having high contrast. In spite of the edge preserving properties of the Nonlocal means filter, due to the high value of filtering parameter, $h = 500$ while smoothing the image, an "overflowing" occurs from the bright region to the dark region across the edge. Processing of such illumination and



Original images

Correction using ground truth values

Correction using white patch assumption

Correction using grey world algorithm

Correction using grey edge algorithm

Correction using Gaussian filter

Correction using bilateral filter

Correction using median filter

Correction using non-local means filter

Figure 12: Example of images from real-world environment corrected after estimating the illumination by different methods and their angular errors.

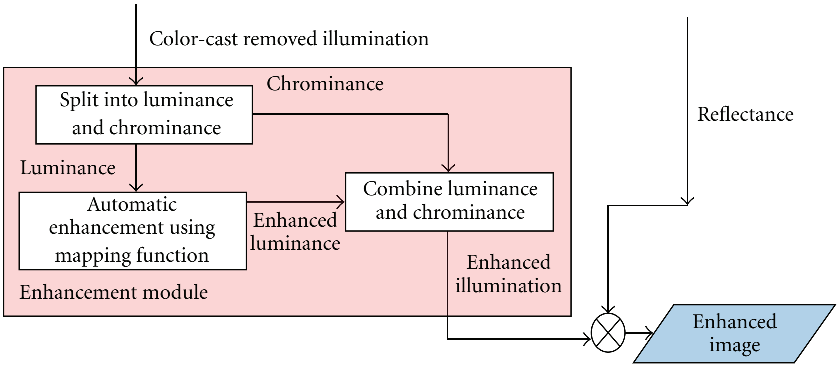

Color-cast removed illumination
Split into luminance and chrominance
Chrominance
Luminance
Automatic enhancement using mapping function
Enhanced luminance
Combine luminance and chrominance
Enhanced illumination
Enhancement module
Reflectance
Enhanced image

Figure 13: Flowchart of the enhancement module.

multiplying it back with the reflectance causes strong halo effects. In order to remove the halo effect, we presegment the image using Mean-shift segmentation algorithm proposed by Comaniciu and Meer [42]. Any segmentation algorithm can be used. The boundaries of the segmented image is used as a preliminary information for the smoothing process. While trying to estimate the illumination, when we consider a neighborhood for every pixel, we also consider the same spatial neighborhood of the presegmented image and if an edge occurs in the presegmented image, less smoothing is done ($h = 0.01$). This helps in preserving high contrast boundaries of the image and thus removes the halo effect from the image.

Once the illumination component $\mathbf{L}(\mathbf{x})$ has been estimated, the reflectance component of the image $\mathbf{R}(\mathbf{x})$ for every pixel $\mathbf{x}$ can be calculated as the pixel-by-pixel ratio of the image and the illumination component and can be expressed as

$$\mathbf{R}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x})}{\mathbf{L}(\mathbf{x})}, \tag{14}$$

where $\mathbf{I}(\mathbf{x})$ is the original image. Alternatively, in the logarithm domain, the difference between $\mathbf{I}(\mathbf{x})$ and $\mathbf{L}(\mathbf{x})$ can be used to estimate $\mathbf{R}(\mathbf{x})$.

As shown in Section 3.2, once we estimate the color of illumination, we remove the effect of color cast from $\mathbf{L}(\mathbf{x})$. We first estimate the color of illumination as shown in (10). Then as shown in (11), we remove the effect of color cast from each of the 3 color channels (red, green and blue) and then combine all 3 new color channels to get color-corrected illumination image, $\mathbf{L}_{cc}(\mathbf{x})$.

4.2. Automatic Enhancement of Illumination. In order to preserve the color properties of the image, we modify only the brightness component of the image. The illumination $\mathbf{L}_{cc}(\mathbf{x})$ obtained from Section 4.1 is converted from RGB color space to HSV color space and only the luminance (V) channel—$\bar{\mathbf{L}}_{cc_V}(\mathbf{x})$ is modified. Alternatively, CIELAB color space could be used and modification of the lightness channel can be done. The chrominance channels are not modified to preserve the color properties of the image.

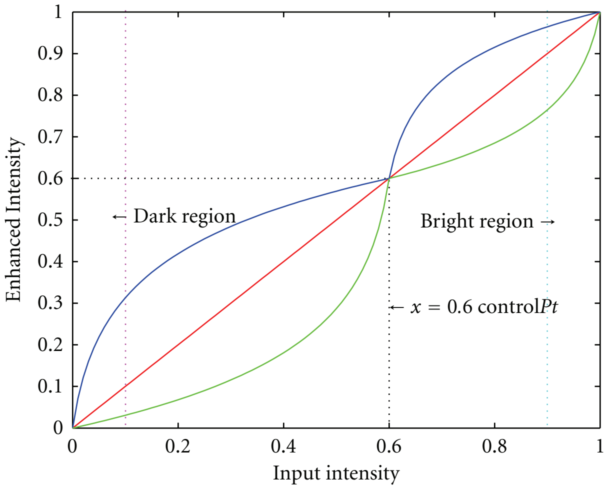


Figure 14: The mapping function for enhancement.

Our method can be used for images that are underexposed, overexposed or a combination of both. Since, we have to deal with different exposure conditions, our enhancement method deals with those cases separately and divide the intensity of the illumination map into 2 different regions as shown in Figure 14.

The division of the region is determined by the positioning of the controlPt. The value of controlPt is computed as follows:

$$\text{control}Pt = \frac{\sum_{\bar{\mathbf{L}}_{cc_V}(\mathbf{x})\le 0.5} 1}{\sum_{\bar{\mathbf{L}}_{cc_V}(\mathbf{x})\le 1} 1}. \tag{15}$$

Intuitively, if an image is heavily underexposed, then a lot of pixels will have very low intensity values. Using (15) we can see that, for such images, the value of control$Pt \rightarrow 1$. Similarly, if an image is heavily overexposed, then a lot of pixels will have very high intensity values causing the value of control$Pt \rightarrow 0$. Thus, we can see that the control$Pt \in [0.5, 1]$ for images that are predominantly underexposed whereas control$Pt \in [0, 0.5]$ for predominantly overexposed images.

To improve the visual quality of images, if the dynamic range of certain parts of the image is increased, some other range must be compressed.

For underexposed regions,

$$\begin{aligned}&\text{Choose blue curve} \quad \text{dark} = \frac{\sum_{\overline{\mathbf{L}}_{cc_V}(\mathbf{x})\leq 0.1} 1}{\sum_{\overline{\mathbf{L}}_{cc_V}(\mathbf{x})\leq 1} 1} > T_1 \\ &\text{Choose green curve} \quad \text{Otherwise.}\end{aligned} \tag{16}$$

If a lot of pixels lie in the dark region, then we have to enhance the values of those pixels by a larger value. As a result, the blue curve will be chosen. Otherwise, we will choose the green curve to enhance the pixels that lie in $(0.1, \text{control}Pt]$ region.

For overexposed regions,

$$\begin{aligned}&\text{Choose green curve} \quad \text{bright} = \frac{\sum_{\overline{\mathbf{L}}_{cc_V}(\mathbf{x})\geq 0.9} 1}{\sum_{\overline{\mathbf{L}}_{cc_V}(\mathbf{x})\leq 1} 1} > T_2 \\ &\text{Choose blue curve} \quad \text{Otherwise.}\end{aligned} \tag{17}$$

Similarly, if a lot of pixels lie in the bright region, then we have to reduce the values of those pixels by a larger value. Therefore, we will choose the green curve. Otherwise, in order to enhance the pixels that lie in $(\text{control}Pt, 0.9)$, we choose the blue curve.

The thresholds $T_1 = 0.01$ and $T_2 = 0.01$ are determined experimentally and those values were chosen that gave the best enhancement results over a wide range of images.

The curves are represented as a logarithmic function. The blue curve can be represented as

$$\overline{\mathbf{L}}_{\text{enh}_V}(\mathbf{x}) = v_1 + \frac{1}{K}\log\left(\left(\overline{\mathbf{L}}_{cc_V}(\mathbf{x}) - v_1\right)\rho + 1\right)(v_2 - v_1), \tag{18}$$

and the green curve can be represented as

$$\overline{\mathbf{L}}_{\text{enh}_V}(\mathbf{x}) = v_2 - \frac{1}{K}\log\left(\left(v_2 - \overline{\mathbf{L}}_{cc_V}(\mathbf{x})\right)\rho + 1\right)(v_2 - v_1). \tag{19}$$

where $K = \log(\rho(v_2 - v_1) + 1$ is a constant and $\overline{\mathbf{L}}_{cc_V}(\mathbf{x}) \in [v_1, v_2]$. For underexposed regions of the image, $\overline{\mathbf{L}}_{cc_V}(\mathbf{x}) \in [0, \text{control}Pt]$ and for overexposed regions, $\overline{\mathbf{L}}_{cc_V}(\mathbf{x}) \in (\text{control}Pt, 1]$. The formulation of the curves in (18) and (19) is inspired by the Weber-Fechner law that states that the relationship between the physical magnitude of the stimuli and the perceived intensity of the stimuli is logarithmic. This relationship was also explored by Stockham [43] and Drago et al. [44] who also recommended a similar logarithmic relationship for tone mapping purposes. Our formulation, inspired by the HVS, is different in flavor from the existing formulations. It takes care of the different exposure conditions simultaneously and automatically estimates the parameters across a wide variety of images, thus enhancing those images without additonal manual intervention.

ρ represents the curvature of the curve, and enhances the intensity of illumination accordingly. For instance, in an underexposed region, due to higher value of *dark*, the intensity has to be increased more and so a higher value of ρ is used for the first curve. Similarly in an overexposed region, due to higher value of *bright*, we decrease the intensity of those regions by a higher value and therefore, a higher value of ρ is used for the second curve. This effect is depicted in Figure 15.

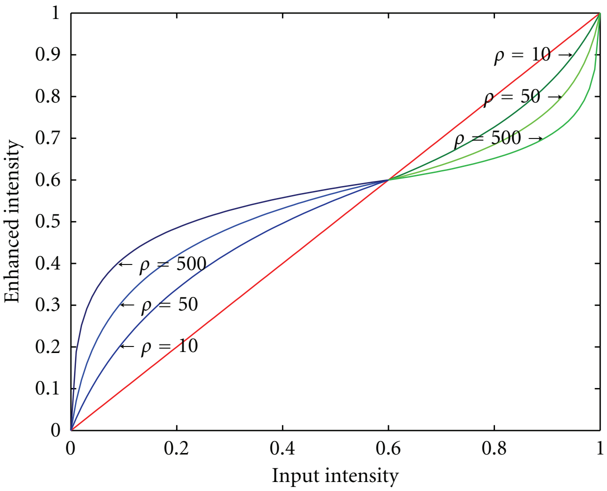


Figure 15: The effects of changing the value of ρ.

$\overline{\mathbf{L}}_{\text{enh}_V}(\mathbf{x})$ is the enhanced luminance. This is combined with the original chrominance to obtain the enhanced illumination—$\mathbf{L}_{\text{enh}}(\mathbf{x})$.

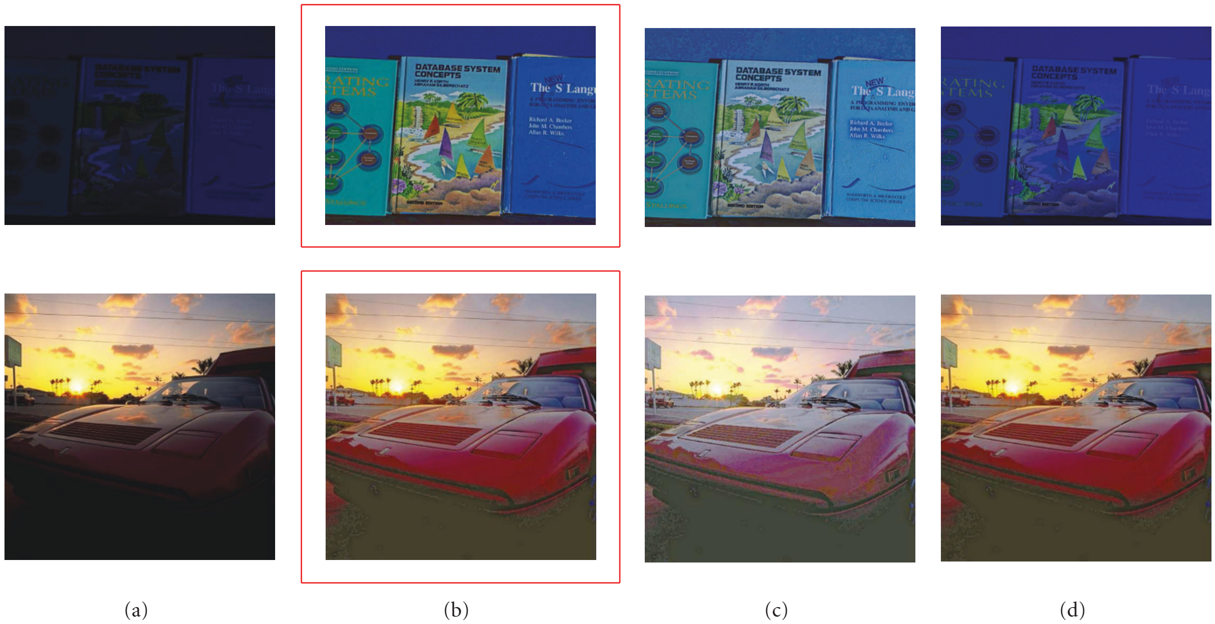
4.3. Combining Enhanced Illumination with Reflectance. The enhanced illumination, $\mathbf{L}_{\text{enh}}(\mathbf{x})$ that was obtained in Section 4.2 is multiplied with the reflectance component $\mathbf{R}(\mathbf{x})$ that was obtained in Section 4.1 to produce the enhanced image $\mathbf{I}_{\text{enh}}(\mathbf{x})$ as follows:

$$\mathbf{I}_{\text{enh}}(\mathbf{x}) = \mathbf{L}_{\text{enh}}(\mathbf{x})\mathbf{R}(\mathbf{x}), \tag{20}$$

The entire process is automated and the enhancement occurs depending on the distribution of the pixels in the image.

5. Results and Discussion

In this section, we consider the computational costs and discuss the results of applying the enhancement algorithm described in Section 4 on a variety of images. We compare our results with histogram equalization [18] and different retinex techniques. We also compare our results with commercial image processing software packages—Picasa, DXO Optics Pro and PhotoFlair. PhotoFlair uses the multiscale retinex with color restoration (MSRCR) algorithm proposed by Jobson et al. [25]. Although we have used many different methods for comparison, we could not use common benchmark image(s) to compare all the methods. This is because, for some methods such as [20–23], and so forth, the source code is not available or for method such as [45], the available source code does not produce desired results or package such as PhotoFlair needs to be purchased. All these techniques use different images for enhancement. So, we use results directly from the respective papers/websites to do comparison with our enhancements. However, as shown in Section 5.5, for a couple of images we use multiple methods.



(a) (b) (c) (d)

Figure 16: Enhancement according to our algorithm. Column (a) is the original images. Column (b) is enhancements according to our implementation. Column (c) is enhancements on all 3 color channels of the illumination. Column (d) is enhancements on the intensity channel of the illumination.

Finally, we perform statistical analysis and quantitative evaluation followed by human validation to demonstrate the effectiveness of our enhancement algorithm.

5.1. Computational Cost. We have implemented the algorithm in MATLAB in Windows XP environment on a PC with Xeon processor. For an image with size 360 × 240 pixels, segmentation takes 6 sec., illumination estimation takes 39 sec. and the enhancement of illumination takes <1 sec. The speed can be improved by implementing the code in C++. Also, faster and more optimized versions of NL means filter can be used to increase the computational efficiency.

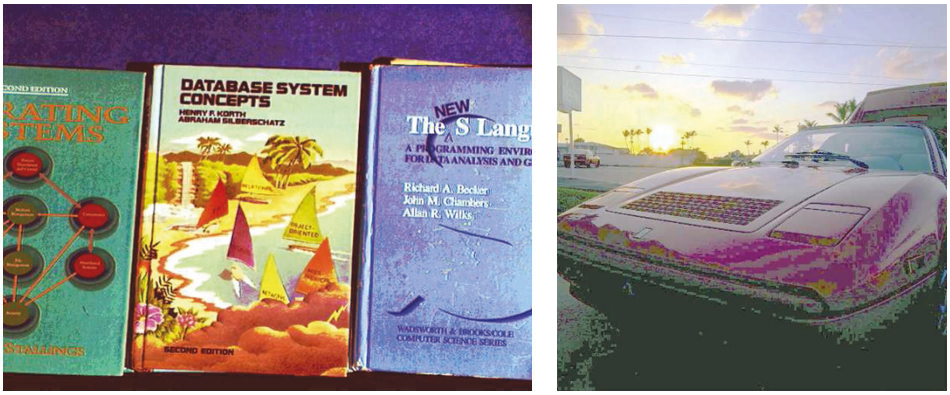
5.2. Enhancement Results. In order to display the results of our enhancement, we consider two different images as shown in Figure 16(a). The top image, obtained from the database provided by Barnard et al. [37], is taken under strong blue illumination while the bottom image is obtained courtesy of P. Greenspun (http://philip.greenspun.com/), and is underexposed.

The results of our algorithm is presented in Figure 16(b). The contrast of the enhanced images is much better than the original images and the color cast from the images has also been removed. For comparison, enhancement was done on all 3 color channels of the original illumination $\mathbf{L}(\mathbf{x})$ as shown in Figure 16(c). It helped in removing the color cast but also resulted in a loss of color from the original image. Enhancement was also done on only the intensity channel of the original illumination $\mathbf{L}(\mathbf{x})$ as shown in Figure 16(d) and it did not help in removing the color cast as shown in the top image. Therefore, our technique produces the best of both worlds—it enhances the color of the scene as well as it helps in removing the color cast from the image. Further, it also gives us a visually pleasant image.

5.3. Comparison with Other Methods. We have also compared our results with existing methods such as histogram equalization. Histogram equalization on all 3 color channels of the original image results in color shift and loss of color as shown in Figure 17. It also results in artifacts. The histogram equalization on only the luminance channel as shown in Figure 18 does not result in color shift but it also does not improve the visual quality of images with color cast.

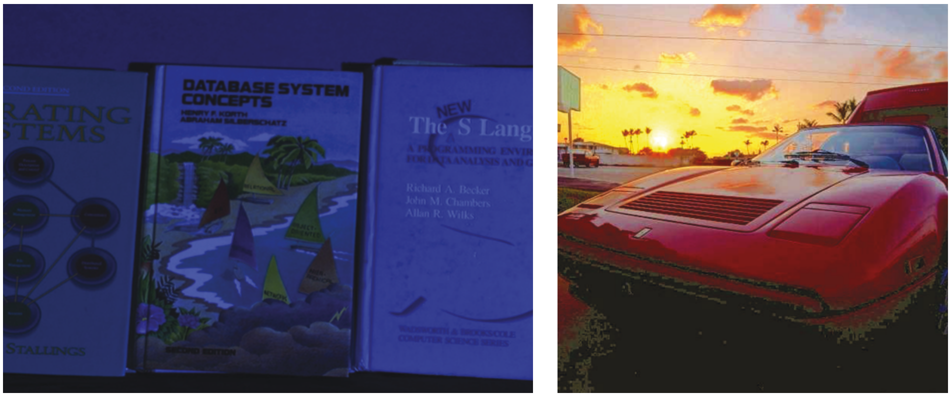
We have compared with the variational approach in retinex proposed by Kimmel et al. [26] and have used our implementation of the proposed technique. This technique uses gamma correction to modify the illumination. The method is not automated and the value of gamma required for enhancing the images depends on the quality of the original images (underexposed/overexposed) and it is cumbersome to manually adjust it. The output from this technique is shown in Figure 19. When we compare this to the output that we get as shown on the top image in Figure 16(b), we can see that our method produces better enhancement.

We also compared with the retinex implementation by McCann in [45] as shown in Figure 20. A disadvantage is the possibility that a halo effect may still exist in the scene as shown in Figure 20 in the title of the rightmost book. The halo effect does not occur in our method because of the prior segmentation of the method and the use of NL-means filter which preserves the edges in a scene.



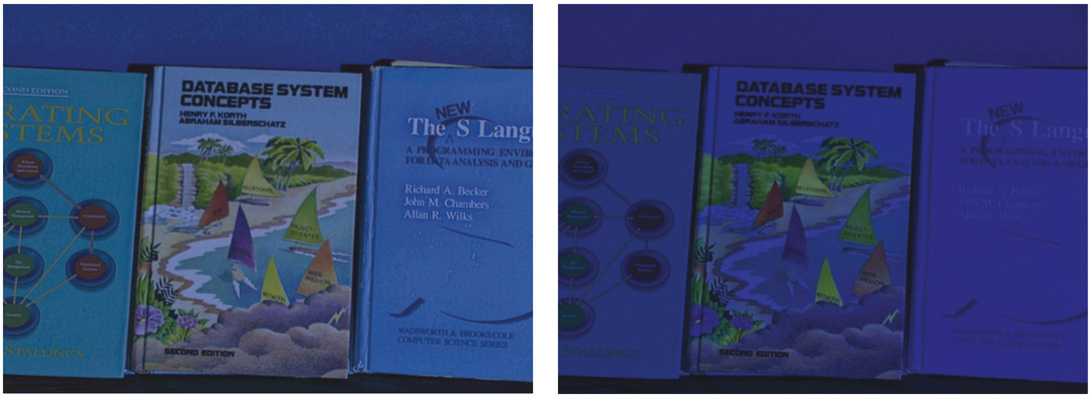
(a) (b)

Figure 17: Histogram equalization on all 3 color channels of the input image. Color shifts and artifacts are quite prominent.



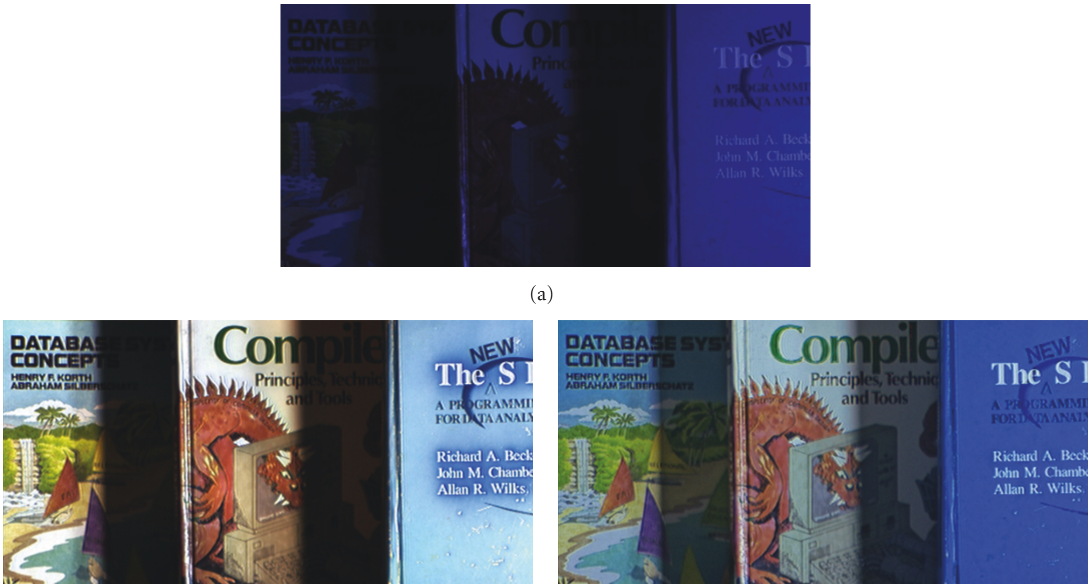
(a) (b)

Figure 18: Histogram equalization on the luminance channel of the input image. Color cast persists and does not improve the contrast of image.



(a) (b)

Figure 19: Enhancement using variational approach of retinex algorithm. The image on the left is enhanced using all 3 channels of the illumination whereas the image on the right is enhanced using only the intensity value of the illumination.



(a)

(b)

(c)

Figure 20: (a) is the original image. (b) is enhanced by McCann retinex and (c) is enhanced by our algorithm. More detail can be seen in the shadows and no halo effect exists in our implementation.

We have also compared our results with a recent techniques proposed by Palma-Amestoy et al. [20] and Provenzi et al. [21–23],and observe that our methods give better contrast and the resulting images look visually better as shown in Figure 21.

We also compared our results with the MSRCR algorithm and observed that our results have a better contrast than the output of the PhotoFlair software as shown in Figure 22. The original images and the output of MSRCR were obtained from NASA website (http://dragon.larc.nasa.gov/retinex/pao/news).

Finally, we compared our algorithm with the output given by Picasa. We used the *Auto Contrast* feature of Picasa and as shown in Figure 23, we can see that we obtain better results. Other parameters can be manually modified in Picasa to get better results but we are building an automated system with no manual tuning of parameters. Even in our approach, if necessary, we can also manually modify parameters to obtain different looking results.

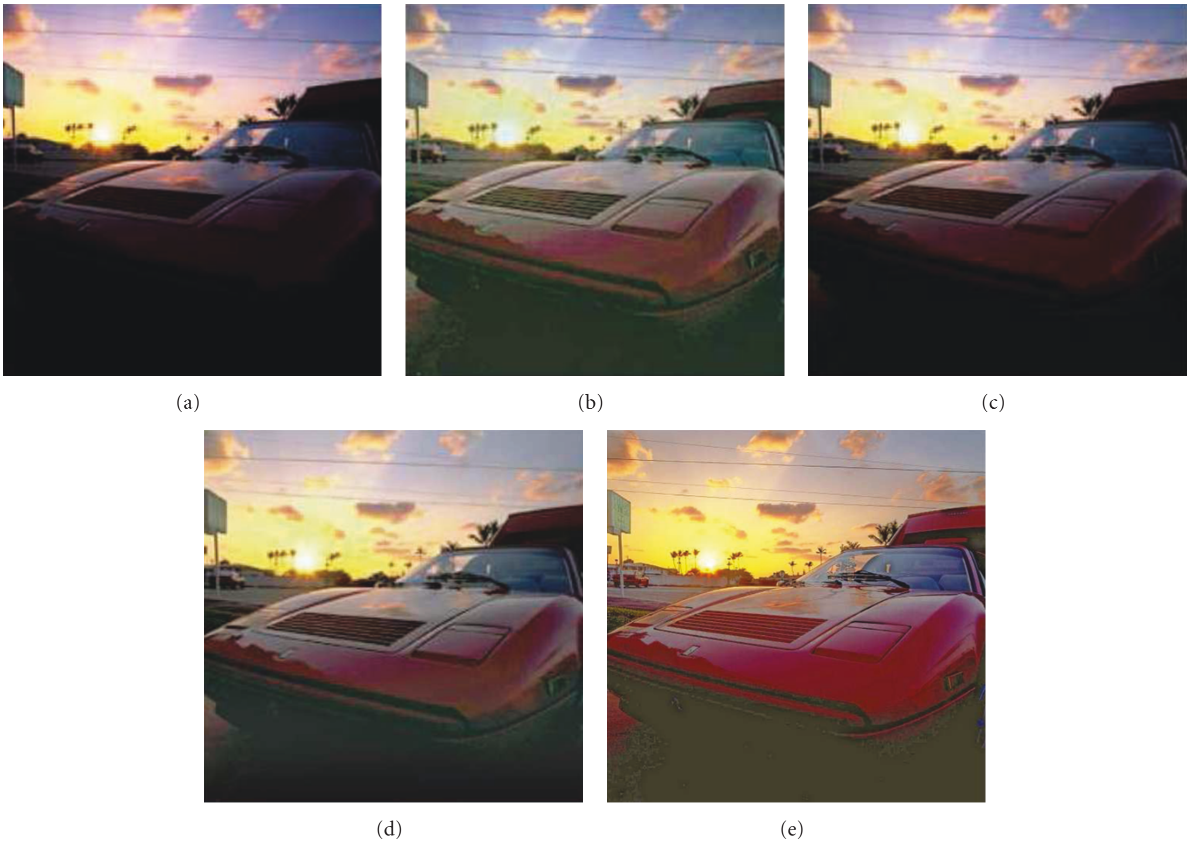
We also compared our algorithm with the output given by DXO Optics Pro and the results are shown in Figure 24.

5.4. Statistical Analysis. A good way to compare an image enhancement algorithm is to measure the change in image quality in terms of brightness and contrast [28]. For measuring the brightness of the image, the mean brightness of the image is considered and for measuring the contrast of the image, local standard deviation of the image (standard deviation of image blocks) is considered. In Figure 25, the mean brightness and local standard deviation of the image blocks for the top image of Figure 16(a) (size of image block is 50×50 pixels) is plotted before and after enhancement. We can see that there is a marked difference (increase) in both the brightness and the local contrast after image enhancement.

5.5. Quantitative Evaluation. A quantitative method to evaluate enhancement algorithms depending on visual representation was devised by Jobson et al. [46] where a region of 2D space of mean of the image and local standard deviation of the image was defined as *visually optimal*, after testing it on a large number of images. We have shown the effects of implementing our algorithm in Figure 26. It can be seen that some of the enhanced images (overexposed and underexposed before enhancement) lie inside the *visually optimal* region whereas other images though do not lie in the *visually optimal* region have a tendency to move towards that region. This happens because those original images are either too dark (very underexposed) or too bright (very overexposed).

We also visualize the effects of other enhancement algorithms in Figure 26. We observe that more images enhanced by our method lie closer/in the *visually optimal* region than the images enhanced by MSRCR. For images that are heavily overexposed, we can see that our enhancements lie closer to the *visually optimal* region than the ones enhanced by Picasa. Some images, for instance, image with $\mu_\mu \approx 42$ and $\mu_\sigma \approx 5$ are enhanced by multiple methods and we can see that our enhanced images lie closer to the *visually optimal* region than the other enhancement methods. On an average, our approach results in "better" enhancement than existing techniques. However, "better" enhancement is very subjective so we conduct human validation of our approach, as described in Section 5.6.

Some interesting observations are as follows—Histogram Equalization (both on 3 color channels or the intensity channel) results in the enhanced image being in the *visually optimal* region whereas our enhanced image lies very close to that region. This is because as can be seen in Figure 17 and Figure 18, a lot of artifacts are generated. These artifacts


(a) (b) (c)

(d) (e)

Figure 21: (a), (b), and (c): Enhancements by Rizzi et al. and Provenzi et al. (a) is enhanced by RSR [22], (b) is enhanced by ACE [21] and (c) is enhanced by RACE [23]. The images are obtained from [23]. (d) and (e): (d) is obtained from [20] whereas (e) is using our algorithm. Note the color contrast amongst the images. Also on zooming in the image, the *prancing horse* can be seen more clearly on (e).

increase the local standard deviation thus causing an increase in the value of μ_σ resulting in the histogram equalized image to lie in the *visually optimal* region. Also for an image that already lies in the *visually optimal* region, we can see that our enhancement does not result in a significant change. In short, our enhancement does not spoil the visual quality of good quality images. This has also been validated by experiments on human observers as shown in Section 5.6.

5.6. Human Validation. To know the effectiveness of our method, we enhanced 40 images using our algorithm. The procedure has 2 steps: (1) We presented the original and the enhanced version simultaneously. The placement of the original and the enhanced version were random(either the left or the right side of the screen) to remove any bias that may exist while selecting an image. We asked each of the 12 human observers independently to rate the image on the "right" side as "*better*", "*same*" or "*worse*" relative to the image on "left" side of the screen and recorded their responses. Scores were thus assigned as 1 for *Better*, 0 for *Same* and -1 for *Worse*. These responses were used to evaluate the "*preference*" for enhanced image. (2) The original image was then presented on the screen and the observers were asked to rate the "*quality*" of the image as "*poor*", "*average*" or "*good*" that was recorded as 1, 2 or 3 respectively. The people were unaware of the fact that the image in Step 2 that is being presented to them is the original image from Step 1 to remove any bias that may occur due to knowledge of the original image.

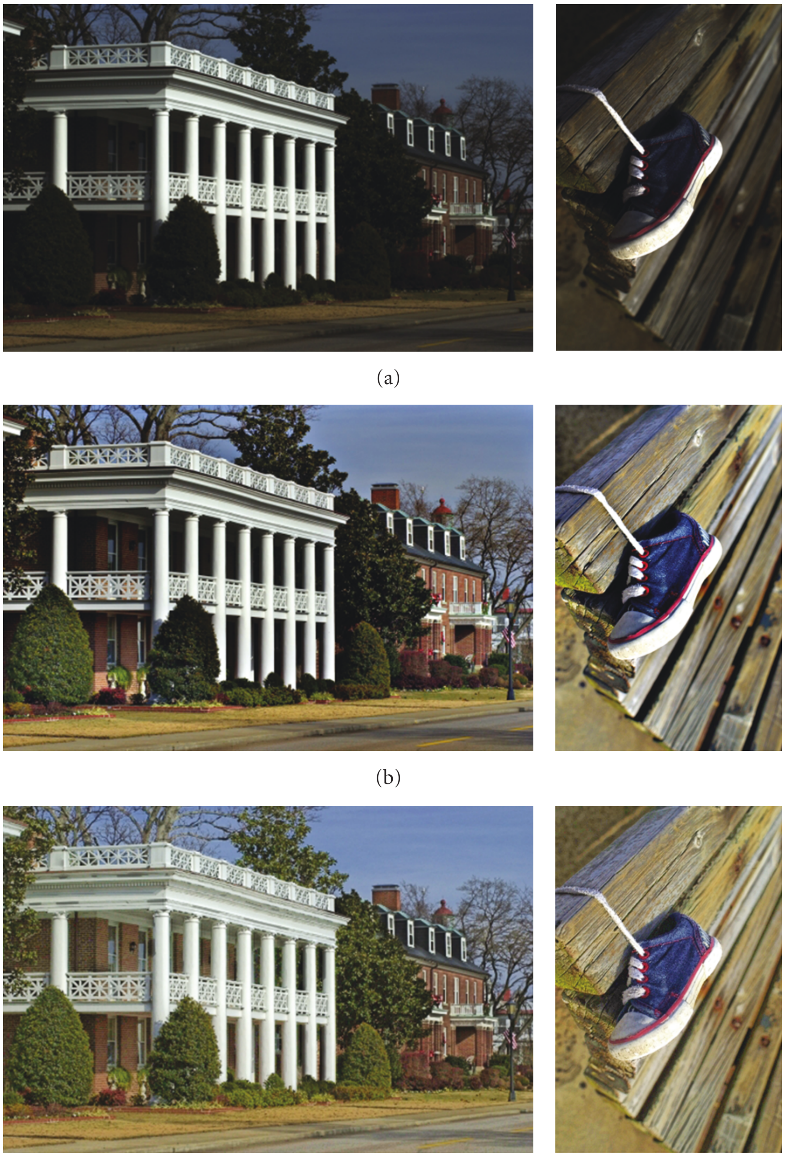
This was repeated for all 40 images in the database. Finally, one image that had the most perceptible difference after enhancement was presented to the subject but with the order flipped from the previous display and the response was noted to check if the subjects were consistent with their responses.

We found that observers were not biased towards selecting any one side of the screen. The responses of all the subjects were also consistent when the "*preference*" ratings for the last comparison was compared with that of its earlier ratings.

Figure 27 shows the original image "*quality*" ratings. We can see that around 72.5% of images were rated *above Average* (*quality* $>= 2$) by people.

Similarly, Figure 28 shows the "*preference*" ratings for the enhancement method. We can see that in most cases, the observers prefer our enhanced images.

As can be seen in Figure 28, for 5 images the observers prefer the original images (*preference* < 0), though the preference is not significantly lower. If we check the image



(a)

(b)

(c)

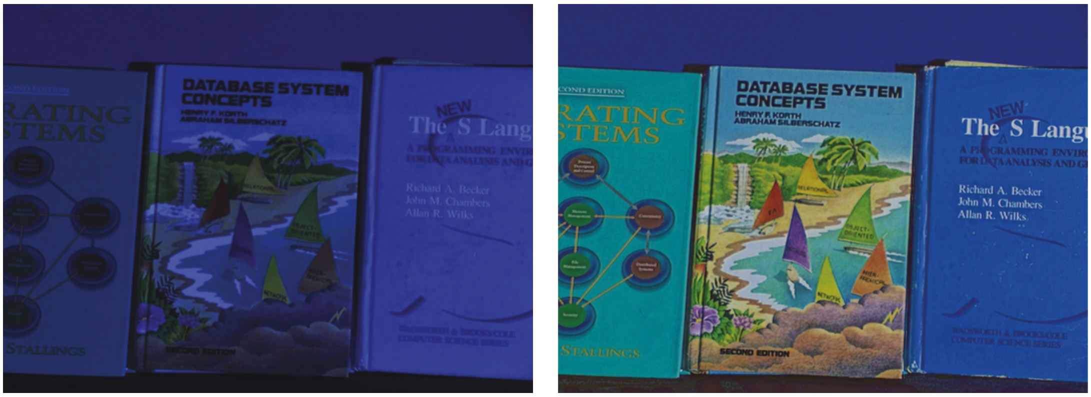
Figure 22: (a) is the original images. (b) is the enhanced images by MSRCR. (c) is the images enhanced by our algorithm. Using our algorithm, we can clearly see the red brick wall within the pillars on the left image and some colored texture on the white part of the shoe can be clearly seen.

quality of those original images (Figure 27), we can see that 3 of those images were rated high (*quality* > 2.5). Also for all the 4 cases where enhanced images were considered to be the same as the original images (*preference* = 0), the quality of the original images was rated was rated high (*quality* > 2.5). For all the cases where the original images were rated *Poor* (*quality* = 1), the observers consistently showed preference for the enhanced images (*preference* > 0).

Further, using Wilcoxon signed-rank test [41], we inferred that the enhanced images are statistically *significantly different* from the original image (The null hypothesis of zero median of the difference is rejected at 5% level) and by comparing the ranks we conclude that the enhanced images have a higher rank than the original images. This implies that people prefer enhanced images over the original images.

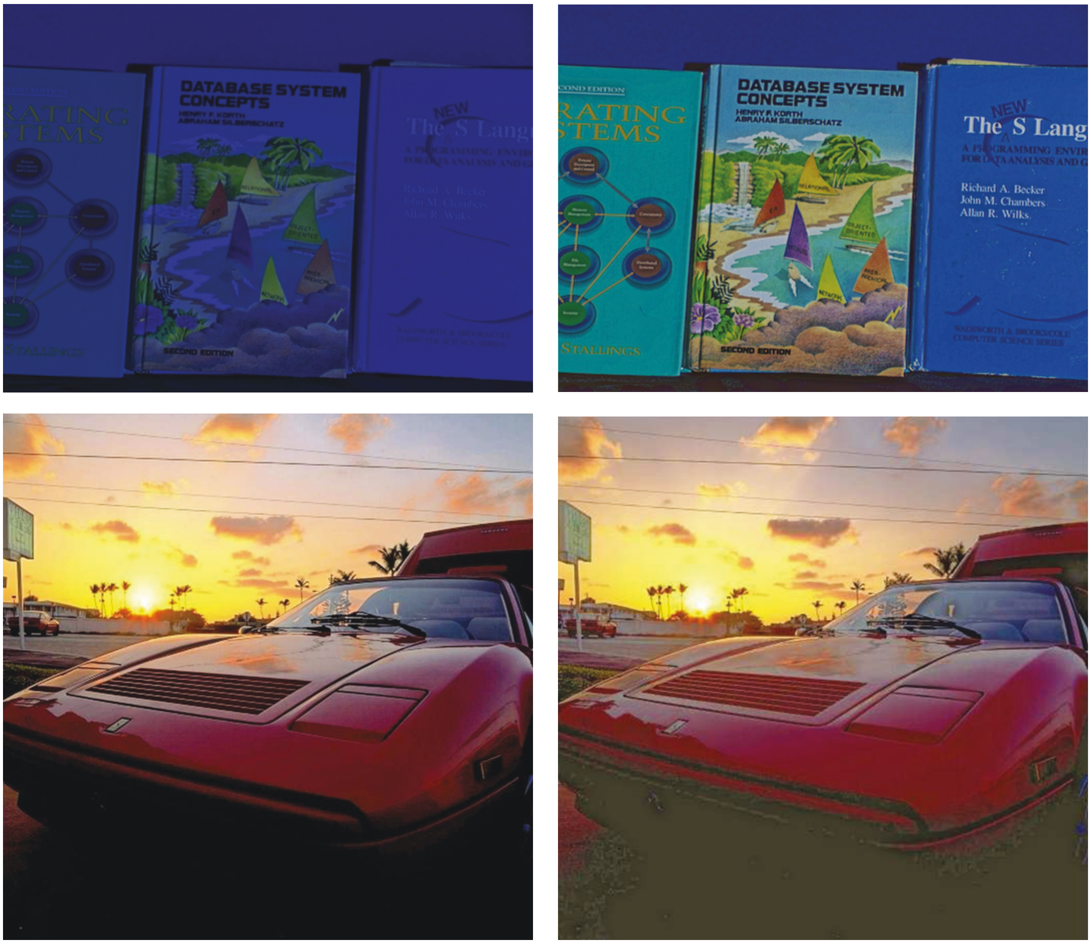
6. Conclusion

We have proposed an automatic enhancement technique for color images that is motivated by human perception and can work well under nonuniform lighting conditions. It provides an unifying framework in which any image can be enhanced automatically irrespective of the inherent color cast in the image or the exposure conditions—both underexposure or overexposure. It thus helps in achieving both *color constancy* and local contrast enhancement in which the human visual system is proficient. Our method estimates the illumination present in the scene and then modifies the illumination to achieve good enhancement depending on some analysis of the distribution of image pixels although, the users can be given control over certain



(a) (b)

Figure 23: (a) is the enhanced output using *Auto Contrast* feature of Picasa and (b) is enhanced using our algorithm.



(a) (b)

Figure 24: (a) is enhanced using DXO Optics Pro and (b) enhanced using our algorithm.

parameters to tune them and adjust the values according to their needs to get customized enhancements. Our method also does not suffer from halo effects during enhancement nor does it suffer from any color shifts or color artifacts or noise magnification. Experimentally, we have compared our enhancement results with results from other existing enhancement techniques and commercial image processing packages and show that our results look "visually" better than the ones produced by existing techniques. Statistically and quantitatively, we have shown that our technique indeed results in enhanced images, and validated our argument by conducting experiments on human observers.

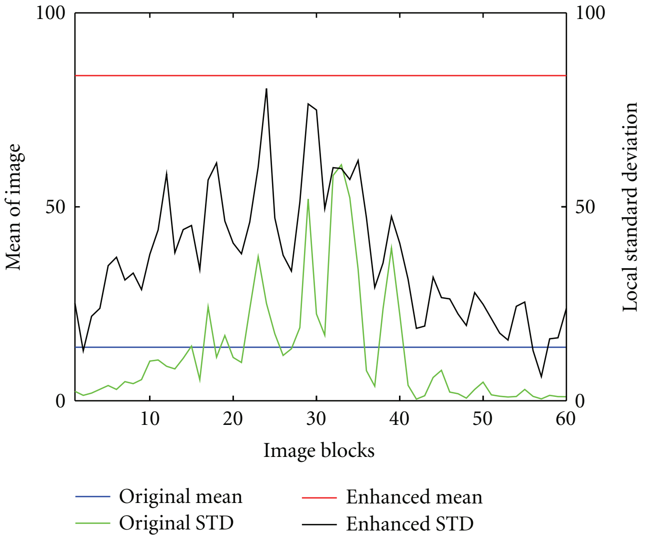


Figure 25: Statistical characteristics of the image before and after enhancement.

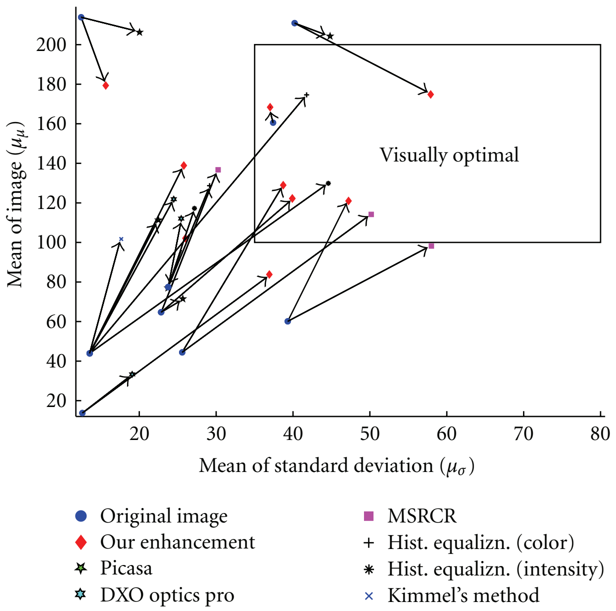


Figure 26: Evaluation of image quality afer image enhancement.

While trying to achieve color constancy, we have used denoising techniques, Gaussian filter, median filter, bilateral filter and Nonlocal means filter. While estimating illumination using the denoising techniques, we observed that the best results were obtained using nonlocal means filter that also best preserves the structure of the image whereas comparatively worse results were obtained with the technique that least preserves the structure of the image. Therefore, intelligent smoothing helps in better estimation of the illuminant value. Experiments on two widely used datasets showed that our techniques give an improvement over existing state-of-the-art color constancy methods.

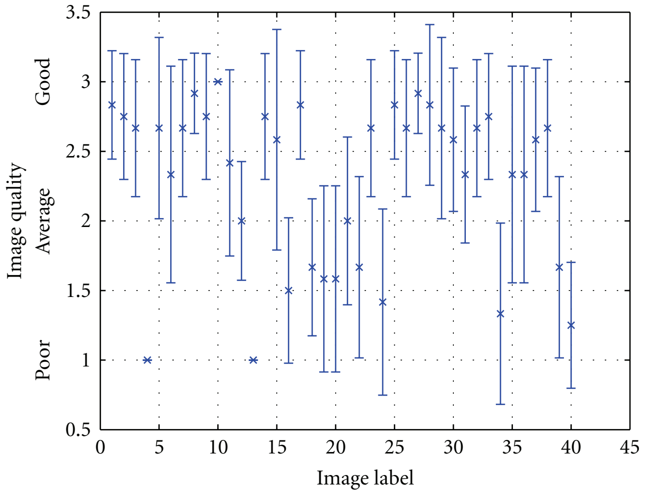


Figure 27: Image *quality* ratings.

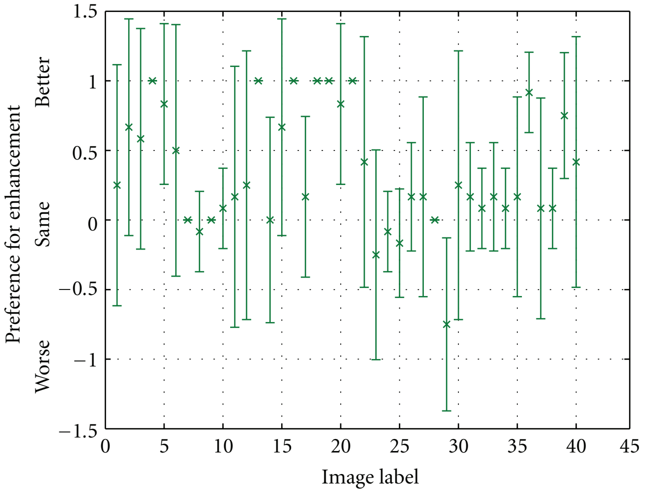


Figure 28: *Preference* for enhancement ratings.

As future work, we would like to ensure that repeated enhancement of the images does not degrade its visual quality. For certain images, post-enhancement though those images look visually better, there is still some information present in the reflectance component that has been left untouched. Enhancing the reflectance component may result in boosting of the texture in the enhanced image that may result in better visual quality of the images. We would also like to organize focus groups to do human validation on a larger dataset.

Regarding color constancy, future work may include ways to combine different approaches in one framework. It may include some amount of learning to know if a particular filter works better for a certain category of images. It will also be interesting to compute the parameters for a filter automatically depending on an image characteristics.

Acknowledgments

This paper was supported by the National Institutes of Health Grant EY016093. This paper is an extended version of the work presented at IEEE International Conference on Image Processing 2009 and IEEE Color and Reflectance in Imaging and Computer Vision Workshop 2009 (in conjunction with ICCV 2009).

References

[1] E. H. Land, "The retinex theory of color vision," *Scientific American*, vol. 237, no. 6, pp. 108–128, 1977.

[2] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, 2007.

[3] D. A. Forsyth, "A novel algorithm for color constancy," *International Journal of Computer Vision*, vol. 5, no. 1, pp. 5–35, 1990.

[4] G. D. Finlayson, S. D. Hordley, and I. Tastl, "Gamut constrained illuminant estimation," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 93–109, 2006.

[5] G. D. Finlayson, S. D. Hordley, and P. M. Hubel, "Color by correlation: a simple, unifying framework for color constancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1209–1221, 2001.

[6] A. Chakrabarti, K. Hirakawa, and T. Zickler, "Color constancy beyond bags of pixels," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–6, June 2008.

[7] V. C. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity by using a neural network," *Journal of the Optical Society of America A*, vol. 19, no. 12, pp. 2374–2386, 2002.

[8] V. Agarwal, A. Gribok, A. Koschan, and M. Abidi, "Estimating illumination chromaticity via kernel regression," in *Proceedings of IEEE International Conference on Image Processing*, pp. 981–984, 2006.

[9] D. H. Brainard and W. T. Freeman, "Bayesian color constancy," *Journal of the Optical Society of America A*, vol. 14, no. 7, pp. 1393–1411, 1997.

[10] C. Rosenberg, T. Minka, and A. Ladsariya, "Bayesian color constancy with non-Gaussian models," in *Proceedings of Neural Information Processing System*, 2003.

[11] G. Buchsbaum, "A spatial processor model for object colour perception," *Journal of the Franklin Institute*, vol. 310, no. 1, pp. 1–26, 1980.

[12] G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Proceedings of the 12th Color Imaging Conference: Color Science and Engineering: Systems, Technologies, Applications*, pp. 37–41, November 2004.

[13] B. Funt, V. Cardei, and K. Barnard, "Committee-based color constancy," in *Proceedings of the 4th IS and T/SID Color Imaging Conference: Color Science, Systems and Applications*, pp. 58–60, Scottsdale, Ariz, USA, November 1996.

[14] A. Gijsenij and T. Gevers, "Color constancy using natural image statistics," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[15] J. van de Weijer, C. Schmid, and J. Verbeek, "Using high-level visual information for color constancy," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, pp. 1–8, October 2007.

[16] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Improving color constancy using indoor-outdoor image classification," *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2381–2392, 2008.

[17] R. Lu, A. Gijsenij, T. Gevers, V. Nedović, D. Xu, and J.-M. Geusebroek, "Color constancy using 3D scene geometry," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 1749–1756, Kyoto, Japan, September-October 2009.

[18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice-Hall, Upper Saddle River, NJ, USA, 3rd edition, 2006.

[19] J.-L. Starck, F. Murtagh, E. J. Candès, and D. L. Donoho, "Gray and color image contrast enhancement by the curvelet transform," *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 706–717, 2003.

[20] R. Palma-Amestoy, E. Provenzi, M. Bertalmío, and V. Caselles, "A perceptually inspired variational framework for color enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 458–474, 2009.

[21] A. Rizzi, C. Gatta, and D. Marini, "From retinex to automatic color equalization: issues in developing a new algorithm for unsupervised color equalization," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 75–84, 2004.

[22] E. Provenzi, M. Fierro, A. Rizzi, L. de Carli, D. Gadia, and D. Marini, "Random spray retinex: a new retinex implementation to investigate the local properties of the model," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 162–171, 2007.

[23] E. Provenzi, C. Gatta, M. Fierro, and A. Rizzi, "A spatially variant white-patch and gray-world method for color image enhancement driven by local contrast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1757–1770, 2008.

[24] M. Bertalmío, V. Caselles, and E. Provenzi, "Issues about retinex theory and contrast enhancement," *International Journal of Computer Vision*, vol. 83, no. 1, pp. 101–119, 2009.

[25] D. J. Jobson, Z.-U. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, 1997.

[26] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel, "A variational framework for retinex," *International Journal of Computer Vision*, vol. 52, no. 1, pp. 7–23, 2003.

[27] E. Provenzi, L. de Carli, A. Rizzi, and D. Marini, "Mathematical definition and analysis of the Retinex algorithm," *Journal of the Optical Society of America A*, vol. 22, no. 12, pp. 2613–2621, 2005.

[28] L. Tao, R. Tompkins, and V. Asari, "An illuminance-reflectance model for nonlinear enhancement of color images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[29] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg, "Multiscale model of adaptation and spatial vision for realistic image display," in *Proceedings of the Annual Conference on Computer Graphics (SIGGRAPH '98)*, pp. 287–298, ACM, July 1998.

[30] G. W. Larson, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 291–306, 1997.

[31] M. Ashikhmin, "A tone mapping algorithm for high contrast images," in *Proceedings of the 13th Eurographics Workshop on Rendering*, pp. 145–155, Eurographics Association, Pisa, Italy, June 2002.

[32] A. Gijsenij and T. Gevers, "Color constancy by local averaging," in *Proceedings of the 14th International Conference on Image Analysis and Processing Workshops (ICIAP '07)*, pp. 171–174, IEEE Computer Society, Modena, Italy, September 2007.

[33] J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, Reading, Mass, USA, 1977.

[34] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the 6th IEEE International Conference on Computer Vision*, pp. 839–846, IEEE Computer Society, Bombay, India, January 1998.

[35] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 123–139, 2008.

[36] S. D. Hordley and G. D. Finlayson, "Re-evaluating colour constancy algorithms," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 76–79, August 2004.

[37] K. Barnard, L. Martin, B. Funt, and A. Coath, "A data set for color research," *Color Research and Application*, vol. 27, no. 3, pp. 147–151, 2002.

[38] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms—part II: experiments with image data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 985–996, 2002.

[39] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[40] F. Ciurea and B. Funt, "A large image database for color constancy research," in *Proceedings of the 11th Color Imaging Conference: Color Science and Engineering:Systems, Technologies, Applications*, pp. 160–164, November 2003.

[41] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[42] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[43] T. Stockham, "Image processing in the context of a visual model," *Proceedings of the IEEE*, vol. 60, no. 7, pp. 828–842, 1972.

[44] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes," *Computer Graphics Forum*, vol. 22, no. 3, pp. 419–426, 2003.

[45] B. Funt, F. Ciurea, and J. McCann, "Retinex in matlab," in *Proceedings of the 8th IS and T/SID Color Imaging Conference: Color Science, Systems and Applications*, pp. 112–121, November 2000.

[46] D. J. Jobson, Z.-U. Rahman, and G. A. Woodell, "The statistics of visual representation," in *Visual Information Processing XI*, vol. 4736 of *Proceedings of SPIE*, pp. 25–35, Orlando, Fla, USA, April 2002.