

Low-light Image Enhancement

Bhavya Vasudeva, Puneesh Deora

Abstract—Images captured under low-light conditions exhibit low visibility. Besides poor aesthetic quality, they may drastically hamper the performance of many computer vision and multimedia algorithms, particularly those that are designed for high-quality inputs. In this paper, we implement a low-light image enhancement method based on illumination map estimation. The captured low-light image is modelled using the retinex model which decomposes the image into the desired image and the illumination map. Firstly, an initial illumination is obtained for each pixel by finding the highest intensity across the R, G, B channels. This illumination map is refined by solving a optimization problem by using the alternating direction method of multipliers. The approach was tested on a number of low-light images and the results show that the technique can be used to recover the hidden information buried in low-light inputs while keeping the visual quality of the image intact.

Index Terms—Low-light image enhancement, illumination, retinex, illumination estimation.

I. INTRODUCTION

High-visibility images provide complete information, minute details of the captured scene. Besides suffering from poor aesthetic quality, images captured under low-light conditions can deteriorate the working of computer vision algorithms as most of them are specifically designed for high quality input images. Consequently there is a great need for enhancing the images captured under poorly lit conditions which may be critically required in the areas such as pedestrian identification, security cameras, object recognition, etc. Histogram equalization based techniques [1]–[3] have often been used for contrast enhancement. Other methods may include gamma correction, which tends to have a significant drawback that it is a point-wise transformation and does not take into account the relationship of a pixel with its neighboring pixels. This may have a drastic impact on the resultant image which is often found to be inconsistent with real-looking scenes. There have been attempts to enhance low-light images based on the Retinex theory [4] of color vision.

Retinex based techniques such as single-scale Retinex [5], multi-scale Retinex [6] treat the reflectance as the final enhanced result, which is often found not to be compatible with realistic scenes and gives over-enhanced results in most cases. There have been other techniques focusing on the refinement of initial estimated illumination [7], [8], with [9] focusing on simultaneous estimation of both reflectance and illumination. Other techniques based on the dehazing algorithms [10] have also been proposed which attempt to dehaze the inverted input low-light images, consequently invert them again to obtain

B. Vasudeva and P. Deora are undergraduate students in the department of Electronics and Communication Engineering at Indian Institute of Technology Roorkee (IITR), Uttarakhand, India.

B. Vasudeva, 16116016, e-mail: bvasudeva@ec.iitr.ac.in
P. Deora, 16116047, e-mail: pdeora@ec.iitr.ac.in

the resultant enhanced image. We attempt to implement the method proposed in [11] which focuses on refining the initial illumination map by solving an optimization problem based on alternating direction method of multipliers (ADMM). In this work we also attempt to enhance the low-light images using the techniques based on histogram equalization.

The rest of the paper is organized as follows: Section II is focused on the methodology, Section III reports the results of the method, Section IV concludes the paper.

II. METHODOLOGY

The approach is based on the Retinex model, where the captured low-light image \mathbf{L} is composed of the desired scene \mathbf{R} and the illumination map \mathbf{T} , as follows:

$$\mathbf{L} = \mathbf{R} \circ \mathbf{T} \quad (1)$$

where \circ denotes the Hadamard product. By estimating the illumination map \mathbf{T} , the enhanced image can be obtained as:

$$\mathbf{R} = \mathbf{L}/(\mathbf{T} + \epsilon) \quad (2)$$

where the division is performed element wise and ϵ is a small positive constant to avoid division by zero and saturation of the result. An initial estimate of the illumination map $\hat{\mathbf{T}}$ for each pixel p , is obtained as per the following operation:

$$\hat{\mathbf{T}}(p) = \max_{c \in \{R, G, B\}} \mathbf{L}^c(p) \quad (3)$$

This approach is adopted to non-uniformly enhance the illumination of the low-light images, rather than boosting the global illumination only. It is based on the principle that the illumination is at least the maximal value of three channels (R, G, B) at a particular pixel location. In order to refine the illumination map, so that the overall structure is preserved and the textural variations are also smoothed, the following optimization problem is solved:

$$\min_{\mathbf{T}} \|\hat{\mathbf{T}} - \mathbf{T}\|_F^2 + \alpha \|\mathbf{W} \circ \nabla \mathbf{T}\|_1 \quad (4)$$

where $\|\cdot\|_F$ is the Frobenius norm, α is the balancing coefficient, \mathbf{W} is the weight matrix, $\|\cdot\|_1$ is the ℓ -1 norm, and ∇ denotes the gradient operator.

A. Solving the Optimization Problem

As the ℓ -1 norm is not differentiable, this problem can be solved by using the ADMM [12]. Since both the terms in the objective function involve \mathbf{T} , an auxiliary variable \mathbf{G} is introduced for making the problem separable. Accordingly, $\nabla \mathbf{T} = \mathbf{G}$ is added as a constraint. As a result, the earlier optimization problem is equivalent to the following:

$$\min_{\mathbf{T}, \mathbf{G}} \|\hat{\mathbf{T}} - \mathbf{T}\|_F^2 + \alpha \|\mathbf{W} \circ \mathbf{G}\|_1 \quad \text{s. t.} \quad \nabla \mathbf{T} = \mathbf{G} \quad (5)$$

The augmented Lagrangian function for this problem can be formulated as:

$$\mathcal{L} = \|\hat{\mathbf{T}} - \mathbf{T}\|_F^2 + \alpha \|\mathbf{W} \circ \mathbf{G}\|_1 + \Phi(\mathbf{Z}, \nabla \mathbf{T} - \mathbf{G}) \quad (6)$$

$\Phi(\mathbf{Z}, \nabla \mathbf{T} - \mathbf{G}) = \frac{\mu}{2} \|\nabla \mathbf{T} - \mathbf{G}\|_F^2 + \langle \mathbf{Z}, \nabla \mathbf{T} - \mathbf{G} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes matrix inner product, μ denotes positive penalty scalar, and \mathbf{Z} is the Lagrangian multiplier. As per the augmented Lagrangian method, an iterative process of updating one variable at a time, keeping the other variables fixed, is followed, and each step has a simple closed-form solution. The three sub-problems are discussed below:

1) *T Sub-Problem*: Collecting the terms involving \mathbf{T} , the \mathbf{T} sub-problem is described as:

$$\mathbf{T}^{(k+1)} \leftarrow \arg \min_{\mathbf{T}} \|\hat{\mathbf{T}} - \mathbf{T}\|_F^2 + \Phi(\mathbf{Z}^{(k)}, \nabla \mathbf{T} - \mathbf{G}^{(k)}) \quad (7)$$

In order to obtain the solution, the above equation is differentiated with respect to \mathbf{T} and set to 0. This leads to:

$$\begin{aligned} 2(\mathbf{T} - \hat{\mathbf{T}}) + \mu^{(k)} \mathbf{D}^T (\mathbf{D}\mathbf{T} - \mathbf{G}^{(k)}) + \mathbf{D}^T \mathbf{Z}^{(k)} &= 0 \\ \Rightarrow (2\mathbf{I} + \mu^{(k)} \mathbf{D}^T \mathbf{D}) \mathbf{T} &= 2\hat{\mathbf{T}} + \mu^{(k)} \mathbf{D}^T (\mathbf{G}^{(k)} - \frac{\mathbf{Z}^{(k)}}{\mu^{(k)}}) \end{aligned} \quad (8)$$

where \mathbf{I} denotes the identity matrix and $(\cdot)^T$ is the transpose operator. The operations $\mathbf{D}\mathbf{x}$ and $\mathbf{D}^T \mathbf{v}$ represent $\text{reshape}(\mathbf{D}\mathbf{x})$ and $\text{reshape}(\mathbf{D}^T \mathbf{v})$, respectively, where $\mathbf{X} \in \mathbb{R}^{M \times N}$, $\mathbf{x} = \text{vec}(\mathbf{X})$, $\mathbf{V} \in \mathbb{R}^{2M \times N}$ or $\mathbb{R}^{M \times 2N}$, $\mathbf{v} = \text{vec}(\mathbf{V})$. The operator $\text{vec}(\cdot)$ vectorizes a matrix and $\text{reshape}(\cdot)$ converts a vector back to the matrix form. The vectorization is performed by stacking the columns one after the other and reshaping is performed accordingly. \mathbf{D} represents the finite differences approximation of the horizontal and vertical gradients [13]. Thus, $\mathbf{D} = [\mathbf{D}_x^T \ \mathbf{D}_y^T]^T$, such that:

$$\begin{aligned} \mathbf{D}\mathbf{x} &= \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{D}_x \mathbf{x} \\ \mathbf{D}_y \mathbf{x} \end{bmatrix} \\ \mathbf{D}^T \mathbf{v} &= \begin{bmatrix} \mathbf{D}_x^T & \mathbf{D}_y^T \end{bmatrix} \begin{bmatrix} \mathbf{v}_a \\ \mathbf{v}_b \end{bmatrix} = \mathbf{D}_x^T \mathbf{v}_a + \mathbf{D}_y^T \mathbf{v}_b \end{aligned} \quad (9)$$

where the vectors \mathbf{v}_a and \mathbf{v}_b contain the first MN entries and the last MN entries of \mathbf{v} , respectively. In order to implement these operations, \mathbf{D}_x and \mathbf{D}_y can be found using:

$$\mathbf{D}_x \mathbf{x} = \text{vec}(d_x * \mathbf{X}), \quad \mathbf{D}_y \mathbf{x} = \text{vec}(d_y * \mathbf{X}) \quad (10)$$

where $*$ denotes convolution operation, and d_x and d_y are the convolution kernels representing discrete forward differences, as follows:

$$d_x = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad d_y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (11)$$

Also, $\mathbf{V}_i = \text{reshape}(\mathbf{v}_i)$ for $i \in \{a, b\}$ and $\mathbf{V}_i \in \mathbb{R}^{M \times N}$, so that:

$$\mathbf{D}_x^T \mathbf{v}_a = \text{vec}(d_x^f * \mathbf{V}_a), \quad \mathbf{D}_y^T \mathbf{v}_b = \text{vec}(d_y^f * \mathbf{V}_b) \quad (12)$$

where d_x^f and d_y^f are the convolution kernels obtained by flipping d_x and d_y (along both x -axis and y -axis), respectively.

The convolution is performed with circular padding, so that \mathbf{D}_x and \mathbf{D}_y are circulant matrices of size $MN \times MN$ each. Simplification of (10) gives:

$$\mathbf{D}_x = \begin{bmatrix} -1 & 0 & \dots & 1 & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & \ddots & 0 & 1 \\ 1 & \ddots & \ddots & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & 0 & 1 & 0 & \dots & -1 \end{bmatrix} \quad (13)$$

$$\mathbf{D}_y = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \\ 1 & 0 & \dots & 0 & -1 \end{bmatrix} \quad (14)$$

Using the property that circular convolution in spatial domain is equivalent to multiplication in discrete Fourier domain, application of 2-D fast Fourier transform (FFT) on (8) leads to:

$$\mathbf{T}^{(k+1)} \leftarrow \mathcal{F}^{-1} \left(\frac{\mathcal{F}(2\hat{\mathbf{T}} + \mu^{(k)} \mathbf{D}^T (\mathbf{G}^{(k)} - \frac{\mathbf{Z}^{(k)}}{\mu^{(k)}}))}{2 + \mu^{(k)} \sum_{i \in \{x,y\}} \overline{\mathcal{F}(d_i^e)} \circ \mathcal{F}(d_i^e)} \right) \quad (15)$$

where $\mathcal{F}(\cdot)$ is the 2-D FFT operator, while $\mathcal{F}^{-1}(\cdot)$ and $\overline{\mathcal{F}(\cdot)}$ stand for the 2-D inverse FFT and the complex conjugate of $\mathcal{F}(\cdot)$, respectively. The division is performed element-wise, so $\mathbf{2}$ is a matrix with all entries as 2 and size $M \times N$ and d_i^e is a matrix of size $M \times N$ obtained by padding d_i with the proper number of zeros.

The second term in the denominator is the result of the following:

$$\mathbf{D}^T \mathbf{D} = \mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y \quad (16)$$

It can be shown that these terms can be obtained by the circular convolution of d_x^f and d_y^f with d_x and d_y , respectively. Also,

$$\begin{aligned} d_i^f * d_i &= d_i \oplus d_i, \quad i \in \{x, y\} \\ \Rightarrow \mathcal{F}(d_i \oplus d_i) &= \overline{\mathcal{F}(d_i)} \circ \mathcal{F}(d_i) \end{aligned} \quad (17)$$

where \oplus denotes the correlation operation, which is also performed with circular padding.

Since \mathbf{D} is very large, it can only be used for smaller images of size $M \times N \sim 100 \times 100$. In order to use \mathbf{D} for images of larger size, the convolution kernels can be put to use. However, the use of matrix multiplication would be less complex for large matrices as compared to convolution with circular padding. Therefore, the operations in (9) are simplified as follows:

$$\begin{aligned} \mathbf{DX} &\equiv \text{reshape} \left(\begin{bmatrix} \text{vec}(\mathbf{X}_H \mathbf{D}_{H1}) \\ \text{vec}(\mathbf{D}_{V1} \mathbf{X}_V) \end{bmatrix} \right) \\ \mathbf{D}^T \mathbf{V} &\equiv \mathbf{V}_H \mathbf{D}_{H2} + \mathbf{D}_{V2} \mathbf{V}_V \end{aligned} \quad (18)$$

where $\mathbf{D}_{H1}, \mathbf{D}_{H2} \in \mathbb{R}^{(N+1) \times N}$ and $\mathbf{D}_{V1}, \mathbf{D}_{V2} \in \mathbb{R}^{M \times (M+1)}$. Also, $\mathbf{D}_{H1} = \mathbf{D}_{V1}^T$ when $M = N$, and $\mathbf{D}_{V2} =$

$-\mathbf{D}_{V1}$. Also, $\mathbf{V}_H = \mathbf{V}_a$ and the rest of the matrices are of the form:

$$\mathbf{D}_{V1} = \begin{bmatrix} -1 & 1 & 0 & \dots & \dots \\ 0 & -1 & 1 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix} \quad (19)$$

$$\mathbf{D}_{H2} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \\ 1 & 0 & \dots & 0 & -1 \end{bmatrix} \quad (20)$$

$$\mathbf{X}_H = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} & x_{11} \\ x_{21} & x_{22} & \dots & x_{2N} & x_{21} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MN} & x_{M1} \end{bmatrix} \quad (21)$$

$$\mathbf{X}_V = \begin{bmatrix} x_{11} & x_{12} & \dots & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ x_{M1} & x_{M2} & \dots & \dots & x_{MN} \\ x_{12} & x_{13} & \dots & x_{1N} & x_{11} \end{bmatrix} \quad (22)$$

$$\mathbf{V}_V = \begin{bmatrix} v_{bMN} & v_{bM1} & v_{bM2} & \dots & v_{bMN-1} \\ v_{b11} & v_{b12} & \dots & \dots & v_{b1N} \\ v_{b21} & v_{b22} & \dots & \dots & v_{b2N} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ v_{bM1} & v_{bM2} & \dots & \dots & v_{bMN} \end{bmatrix} \quad (23)$$

where x_{ij} are the elements of \mathbf{X} and v_{bij} are the elements of \mathbf{V}_b . Thus, (15) can be implemented using (II-A1) and the matrices listed above.

2) *G Sub-Problem*: From (6), dropping the terms which are not related to \mathbf{G} leads to the following G sub-problem:

$$\mathbf{G}^{(k+1)} \leftarrow \arg \min_{\mathbf{G}} \|\mathbf{W} \circ \mathbf{G}\|_1 + \Phi(\mathbf{Z}^{(k)}, \nabla \mathbf{T}^{(k+1)} - \mathbf{G}) \quad (24)$$

The solution to this can be obtained by performing the shrinkage operation as follows:

$$\mathbf{G}^{(k+1)} = \mathcal{S}_{\frac{\alpha \mathbf{W}}{\mu^{(k)}}} \left[\nabla \mathbf{T}^{(k+1)} + \frac{\mathbf{Z}^{(k)}}{\mu^{(k)}} \right] \quad (25)$$

where $\nabla \mathbf{X} \equiv \mathbf{D} \mathbf{X}$ and $\mathcal{S}_{E>0}[\cdot]$ represents the shrinkage operator. It is defined on scalars as:

$$\mathcal{S}_e[a] = \text{sgn}(a) \max(|a| - e, 0) \quad (26)$$

It can be extended to vectors and matrices by performing element-wise operations, i. e. $\mathcal{S}_{\mathbf{E}}[\mathbf{A}]$ performs the shrinkage on the elements of \mathbf{A} with thresholds given by the corresponding entries of \mathbf{E} .

3) *Z and μ Sub-Problem*: The following equations are followed for updating \mathbf{Z} and μ :

$$\begin{aligned} \mathbf{Z}^{(k+1)} &\leftarrow \mathbf{Z}^{(k)} + \mu^{(k)} (\nabla \mathbf{T}^{(k+1)} - \mathbf{G}^{(k+1)}) \\ \mu^{(k+1)} &\leftarrow \mu^{(k)} \rho, \quad \rho > 1 \end{aligned} \quad (27)$$

B. Weighting Strategies

For the structure-aware refinement on the initial illumination map, two possible weighing strategies can be used for the design of \mathbf{W} , such that $\mathbf{W} = \text{reshape}(\mathbf{w})$, where $\mathbf{w} = [\mathbf{W}_x^T \quad \mathbf{W}_y^T]^T$. The first strategy is:

$$\mathbf{W}_x(p) \leftarrow 1, \quad \mathbf{W}_y(p) \leftarrow 1 \quad (28)$$

The second strategy involves the use of relative total variation, so for each location, the weight is set as:

$$\mathbf{W}_x(p) \leftarrow \sum_{q \in \Omega(p)} \frac{G_\sigma(p, q)}{|\sum_{q \in \Omega(p)} G_\sigma(p, q) \nabla_x \hat{\mathbf{T}}(q)| + \epsilon} \quad (29)$$

$$\mathbf{W}_y(p) \leftarrow \sum_{q \in \Omega(p)} \frac{G_\sigma(p, q)}{|\sum_{q \in \Omega(p)} G_\sigma(p, q) \nabla_y \hat{\mathbf{T}}(q)| + \epsilon}$$

where p, q denote two pixels, $\Omega(p)$ denotes the region around pixel p and depends on the size of the kernel. Also,

$$\nabla \mathbf{X} = \begin{bmatrix} \nabla_x \mathbf{X} \\ \nabla_y \mathbf{X} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{D}_x \mathbf{X} \\ \mathbf{D}_y \mathbf{X} \end{bmatrix} \quad (30)$$

$G_\sigma(p, q)$ denotes the Gaussian kernel having standard deviation σ , given by:

$$G_\sigma(p, q) \propto \exp \left(-\frac{\|\mathbf{p} - \mathbf{q}\|^2}{2\sigma^2} \right) \quad (31)$$

Since \mathbf{W} is constructed based on the given $\hat{\mathbf{T}}$ instead of being iteratively updated according to \mathbf{T} , it only needs to be calculated once.

C. Post-Processing

After obtaining the refined illumination map \mathbf{T} , gamma correction is applied on \mathbf{T} , in order to improve the visual perception of the result, as follows:

$$\mathbf{T} \leftarrow \mathbf{T}^\gamma, \quad \gamma < 1 \quad (32)$$

It is noted that in the process of enhancing the image, noise is also enhanced, especially for the very low-light regions. Denoising techniques are required to further improve the visual quality. Two of them are explored:

1) *Block-matching and 3-D filtering (BM3D)* [14]: In this method, the first step is the grouping of regions (2-D blocks of same size) of the image based on their similarity with respect to a reference block. These are stacked together to form a 3-D array on which a linear transform is applied. Denoising is performed by a transform-domain shrinkage such as Wiener filtering, after which this transform is inverted to reproduce the denoised blocks, which can be replaced at their original locations in the image. The overlapping blocks are weight-averaged before replacement. BM3D is used as an off-the-shelf denoising tool. In order to reduce the computational complexity, the image is converted from RGB colorspace to the YUV colorspace and BM3D is applied only on the Y channel. The denoised Y channel is recomposed with the UV channels and the result is converted back to the RGB colorspace. In order to compensate for the non-uniformity in the result, since the darker regions are denoised at the

cost of smoothening of the brighter regions, the following recomposing operation is performed:

$$\mathbf{R}_f \leftarrow \mathbf{R} \circ \mathbf{T} + \mathbf{R}_d \circ (\mathbf{1} - \mathbf{T}) \quad (33)$$

where \mathbf{R}_d and \mathbf{R}_f are the results after denoising and recomposing, respectively.

2) *Bilateral Filtering [15]*: A bilateral filter (BF) is a non-linear, edge-preserving, smoothing filter, defined as:

$$\mathbf{I}^f(p) \leftarrow \frac{\sum_{q \in \Omega(p)} \mathbf{I}(q) f_r(\mathbf{I}(p), \mathbf{I}(q)) g_s(p, q)}{\sum_{q \in \Omega(p)} f_r(\mathbf{I}(p), \mathbf{I}(q)) g_s(p, q)} \quad (34)$$

where \mathbf{I}^f is the filtered image, \mathbf{I} is the original input image, $f_r(p, q)$ is the range kernel to smooth the differences in intensities, and $g_s(p, q)$ is the spatial kernel to smooth the differences in co-ordinates. A BF replaces the intensity at each location with a weighted average of intensity values, such that higher weights are assigned to the values in the nearby region as well as in the closer range with respect to the center of the kernels. In this work, $f_r(p, q) = G_{\sigma_r}(p, q)$ and $g_s(p, q) = G_{\sigma_s}(p, q)$. The extent of smoothing in both the cases varies directly with the length and standard deviation of the respective kernels.

The steps for the whole procedure described above are summarized in Algorithm 1.

Algorithm 1: Low-light image enhancement

Input : Low-light image \mathbf{L} , positive coefficients α, γ, ρ , and $\mu^{(0)}$

Initialize: $k = 0, \mathbf{G}^{(0)} = \mathbf{Z}^{(0)} = \mathbf{0} \in \mathbb{R}^{2M \times N}$

Estimate the initial illumination map $\hat{\mathbf{T}}$ on \mathbf{L} using (3);

while $k < k_0$ **do**

Update $\mathbf{T}^{(k+1)}$ using (15) ;
Update $\mathbf{G}^{(k+1)}$ using (25) ;
Update $\mathbf{Z}^{(k+1)}$ and $\mu^{(k+1)}$ using (27);
 $k = k + 1$;

end

Apply Gamma correction on $\mathbf{T}^{(k_0)}$ using (32) ;

Obtain \mathbf{R} using Gamma corrected $\mathbf{T}^{(k_0)}$ via (2) ;

Denoise the result using BM3D via (33) or using BF via (34) ;

Output : Final enhanced result

III. RESULTS AND DISCUSSION

The values for the parameters ρ , α , and μ for the solver and σ_s and σ_r for the BF were tuned for best possible outputs using an approach based on grid search. In this work, γ , k_0 and σ are kept as 0.8, 50 and 2, respectively.

Fig. 1 shows the effect of changing ρ . It is observed that ρ is a highly sensitive parameter and even a slight variation in its value has a significant impact on the enhanced output. Fig. 2 shows the variation of α on the enhanced result. It can be seen that for Figs. 2 (c)-2 (f), the output image starts to deteriorate, with the appearance of black and blue boundaries on the bottom and top edges, respectively.

The variation of μ can be seen in Fig. 3. It is evident that the quality of final enhanced results starts to degrade when μ lies in [3 - 300] with flower-like structures starting to appear in the resultant output.



Fig. 1. Effect of parameter ρ on the final enhanced result. From (a) to (f) $\rho \in [1, 1.05, 1.10, 1.15, 1.20, 1.25]$.



Fig. 2. Effect of parameter α on the final enhanced result. From (a) to (f) $\alpha \in [0.008, 0.08, 0.8, 8.0, 80.0, 800.0]$.



Fig. 3. Effect of parameter μ on the final enhanced result. From (a) to (f) $\mu \in [0.003, 0.03, 0.3, 3.0, 30.0, 300.0]$

It is important to note that the the values for ρ , α , and μ are required to lie in a certain range in order to obtain a good enhanced output. This range of values varies with the input image. In this work, the value of α is kept as 0.08.

The effect of changing σ_s and σ_r can be seen on the final enhanced result in Fig. 4 and Fig. 5, respectively. As expected, with increase in the value of σ_s more noise can be handled at the cost of blurring in the resultant image, since the weights assigned to intensities at farther locations increase. Also, a large change in σ_r can only lead to a small change in the output. It is because the weights assigned to intensity values

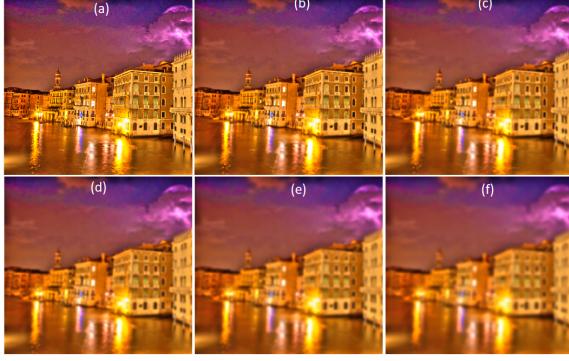


Fig. 4. Effect of parameter σ_s on the result obtained from bilateral filtering. From (a) to (f) $\sigma_s \in [0.5, 1.5, 2.50, 3.5, 4.50, 5.5]$.



Fig. 5. Effect of parameter σ_r on the result obtained from bilateral filtering. From (a) to (f) $\sigma_r \in [0.001, 0.01, 0.1, 1.0, 10.0, 100]$.

which are farther from the reference intensity value increases very slowly.

Fig. 6 illustrates the comparison between BF and BM3D for denoising purposes. It can be seen BM3D performs gives a better result than bilateral filtering. However, BM3D is comparatively more computationally expensive. Therefore, in this work all the results have been reported using BF as it gives satisfactory results in almost all the cases.

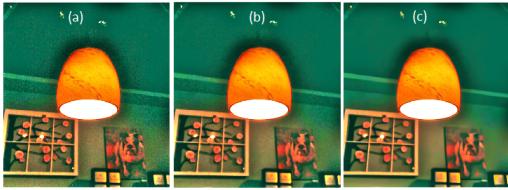


Fig. 6. Enhanced result: (a) without denoising (b) denoising with bilateral filtering (c) denoising with BM3D.

Fig. 7 shows the results of this method on low-light images provided by [11], where the first column shows the low-light input images, second column demonstrates the initial illumination map, third column illustrates the refined illumination map, fourth and fifth column illustrate the enhanced results and the enhanced results obtained after denoising via bilateral filtering, respectively.

Fig. 8 shows the results obtained from different techniques based on histogram equalization, namely *hstran*, *hssep*, *hshsv*, *hsyuv*, and *hsycbcr* on the low-light images in Figs.



Fig. 7. **First column:** low-light images, **second column:** heat map of initial illumination map, **third column:** heat map of estimated illumination map, **fourth column:** enhanced results, **fifth column:** denoised results via bilateral filtering (final enhanced result).

8 (1.(a)) and 8 (2.(a)). For *hstran*, the first step is to obtain the average histogram of the R, G, and B channels. Histogram equalization is applied on this average histogram. The transformation obtained from this process is applied on the R, G, and B channels, which gives the enhanced results in Figs. 8 (1.(b)) and 8 (2.(b)). The results for *hssep* are given in Figs. 8 (1.(c)) and 8 (2.(c)). In this technique, the image is decomposed into R, G, B channels and histogram equalization is applied separately on the three channels. The resultant image is obtained after recomposing the outputs for the three channels. In *hshsv*, the RGB image is converted to HSV color space, where histogram equalization is applied to the V channel. This HSV image is converted back to the RGB space which is the required output, as shown in Figs. 8 (1.(d)) and 8 (2.(d)). The RGB image can also be converted to YUV or YCbCr color space, where histogram equalization is applied on the Y channel, and the obtained image is consequently converted to RGB color space. The results obtained using the *hsyuv* and *hsycbcr* methods are shown in Figs. 8 (1.(e)) and 8 (2.(e)) and Figs. 8 (1.(f)) and 8 (2.(f)), respectively.

Fig. 9 shows some low-light JPG images captured in IITR

campus using Redmi Note 4 and the enhanced images obtained from our implementation. All the images are sub-sampled by a factor of 0.25. Though some granularity and noise is present in the resultant images, the visibility of the content in the images is significantly improved.



Fig. 8. 1.(a), 2.(a) are the low-light images. 1.(b), 2.(b) are the enhanced results obtained from *hstran*, 1.(c), 2.(c) are obtained from without *hssep*, 1.(d), 2.(d) obtained from *hshsv*, 1.(e), 2.(e) obtained from *hsyuv*, 1.(f), 2.(f) obtained from *hsycbr*.

Fig. 10 shows a few images from the ExDark (Exclusively Dark) dataset [16] and the results obtained after low-light image enhancement from our implementation.

Finally, Fig. 11 shows the comparison of results obtained from the LIME sped-up solver [11] and our implementation of the exact solver. It is visible that the results are comparable and very similar for most of the cases.

IV. CONCLUSION

In this paper, the implementation of low-light image enhancement via illumination map estimation is presented. The reported results show that our implementation achieves results which closely resemble the ones reported in the original paper. We have tried to give a clearer explanation for the topics involved in the implementation, have also attempted to compare the results obtained for the aforementioned method with techniques based on histogram equalization. Future work

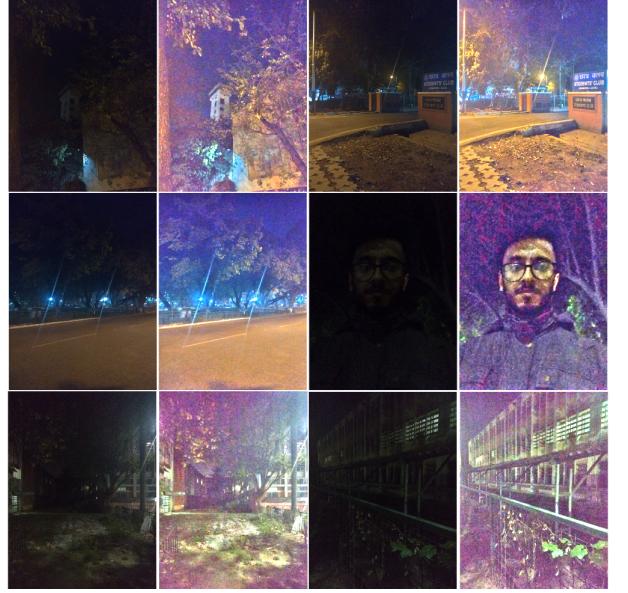


Fig. 9. Some low-light JPG images captured in IITR campus and results obtained from our implementation.



Fig. 10. Some low-light images from ExDark dataset [16] and results obtained from our implementation.



Fig. 11. **First row:** low-light images, **second row:** results of LIME sped-up solver [11], **third row:** results of our implementation of exact solver.

may include developing a method to enhance JPEG low-light images which attempts to suppress the blocking effect seen in low-light enhanced results.

ACKNOWLEDGMENT

The authors would like to thank Dr. Saumik Bhattacharya for his guidance, the help he offered on numerous occasions

by clearing their doubts on a regular basis, and for motivating them throughout the course of this work.

REFERENCES

- [1] E. D. Pisano, S. Zong, B. M. Hemminger, M. DeLuca, R. E. Johnston, K. Muller, M. P. Braeuning, and S. M. Pizer, "Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms," *Journal of digital imaging*, vol. 11, no. 4, p. 193200, 1998.
- [2] H. Cheng and X. Shi, "A simple and effective histogram equalization approach to image enhancement," *Digital Signal Processing*, vol. 14, no. 2, pp. 158 – 170, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S105120040300037X>
- [3] M. Abdullah-Al-Wadud, M. H. Kabir, M. A. Akber Dewan, and O. Chae, "A dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 593–600, May 2007.
- [4] E. H. Land, "The retinex theory of color vision scientific american," 2009.
- [5] D. J. Jobson, Z. Rahman, and G. A. Woodell, "Properties and performance of a center/surround retinex," *IEEE Transactions on Image Processing*, vol. 6, no. 3, pp. 451–462, March 1997.
- [6] D. J. Jobson, Z. 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, July 1997.
- [7] S. Wang, J. Zheng, H. Hu, and B. Li, "Naturalness preserved enhancement algorithm for non-uniform illumination images," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3538–3548, Sep. 2013.
- [8] X. Fu, D. Zeng, Y. Huang, Y. Liao, X. Ding, and J. Paisley, "A fusion-based enhancing method for weakly illuminated images," *Signal Processing*, vol. 129, pp. 82 – 96, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168416300949>
- [9] X. Fu, D. Zeng, Y. Huang, X. Zhang, and X. Ding, "A weighted variational model for simultaneous reflectance and illumination estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2782–2790.
- [10] X. Dong, G. Wang, Y. Pang, W. Li, J. Wen, W. Meng, and Y. Lu, "Fast efficient algorithm for enhancement of low lighting video," in *2011 IEEE International Conference on Multimedia and Expo*, July 2011, pp. 1–6.
- [11] X. Guo, Y. Li, and H. Ling, "Lime: Low-light image enhancement via illumination map estimation," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 982–993, Feb 2017.
- [12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: <http://dx.doi.org/10.1561/2200000016>
- [13] "EE 367 / CS 448I Computational Imaging and Display lecture 6 notes," http://stanford.edu/class/ee367/reading/lecture6_notes.pdf, accessed: 2019-04-07.
- [14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 16, pp. 2080–95, 09 2007.
- [15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–. [Online]. Available: <http://dl.acm.org/citation.cfm?id=938978.939190>
- [16] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset," *Computer Vision and Image Understanding*, vol. 178, pp. 30–42, 2019.