LIME: LOW-LIGHT IMAGE ENHANCEMENT VIA ILLUMINATION MAP ESTIMATION

Kolla Ananta Raj, Manav Nitin Kapadnis, Modi Omkar

Roll No.: 19EE38009, 19EE38010, 19EE38015
Department Of Electrical Engineering
IIT Kharagpur

ABSTRACT

Digital photographs are being employed in a variety of applications. It is employed in nearly every aspect of our lives and technology. Picture enhancement is very important in improving the quality of an image. In addition to decreasing the visual aesthetics of images, this low quality may significantly reduce the performance of many computer vision and multimodal algorithms that are primarily intended for high-quality inputs. Thus, improving the picture in low light is a vital task. In this project, we implement a paper titled - "LIME: Low Light Image Enhancement via Illumination Map Estimation". The Retinex model is used to breakdown the acquired low-light picture into the desired image and the illumination map. To begin, an initial illumination for each pixel is produced by locating the maximum intensity across the R, G, and B channels. This illumination map is then enhanced by solving an optimization problem with the alternating direction multiplier approach. The strategy was tested on a variety of low-light photos, and the findings demonstrate that the technology may be used to retrieve the hidden information buried in low-light inputs while maintaining the image's visual quality. All our codes to generate reproducible results of the paper are available on Github[1].

Index Terms— Illumination Map estimation, Illumination transmission, Low light image enhancement.

1. INTRODUCTION

High-visibility photos give detailed information about the photographed scene. In addition to having poor aesthetic quality, photographs acquired in low-light situations might impair the performance of computer vision algorithms, as most of them are intended for high-quality input photos. As a result, there is a tremendous demand for improving photos acquired in low-light circumstances, which may be crucial in fields such as pedestrian identification, security cameras, object recognition, and so on. Furthermore, techniques based on histogram equalization [1, 2, 3] have frequently been utilized for contrast enhancement. Other solutions may include gamma correction, which has the substantial disadvantage of being a point-wise transformation that does not take into consideration a pixel's interaction with its adjacent pixels. This can have a significant influence on the final image, which is frequently found to be incongruous with realistic-looking settings. Attempts have been made to improve low-light pictures using the Retinex theory of color vision [4].

Retinex-based approaches, such as single-scale Retinex [5] and multi-scale Retinex [6], regard reflectance as the ultimate enhanced result, which is frequently found to be incompatible with realistic settings and produces over-enhanced outcomes in most circumstances. Other strategies have focused on refining initial estimated illumination [7, 8], with [9] focussing on simultaneous estimation of both reflectance and illumination. Other approaches based on dehazing algorithms [10] have also been developed, in which the inverted input low-light photos are dehazed and then inverted again to create the final improved image. We try to apply the strategy provided in [11], which focuses on improving the initial illumination map by solving an optimization problem using the alternating direction method of multipliers (ADMM). In this study, we aim to improve low-light photos using histogram equalization approaches.

The rest of the paper is organized as follows: Section 2 is focused on the methodology, and Section 3 concludes the paper.

2. METHODOLOGY

The approach is based on the Retinex model[12], where the captured low-light image **L** is composed of the desired image **R** and the illumination map **T**, as follows:

$$\mathbf{L} = \mathbf{R} \circ \mathbf{T} \tag{1}$$

Where the operator $\circ$ means element wise multiplication. By estimating the illumination map **T**, the enhanced image can be obtained as:

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

Where the division is performed element wise and very small constant ϵ is added in denominator to avoid division

[1] https://github.com/rajanant49/LIME

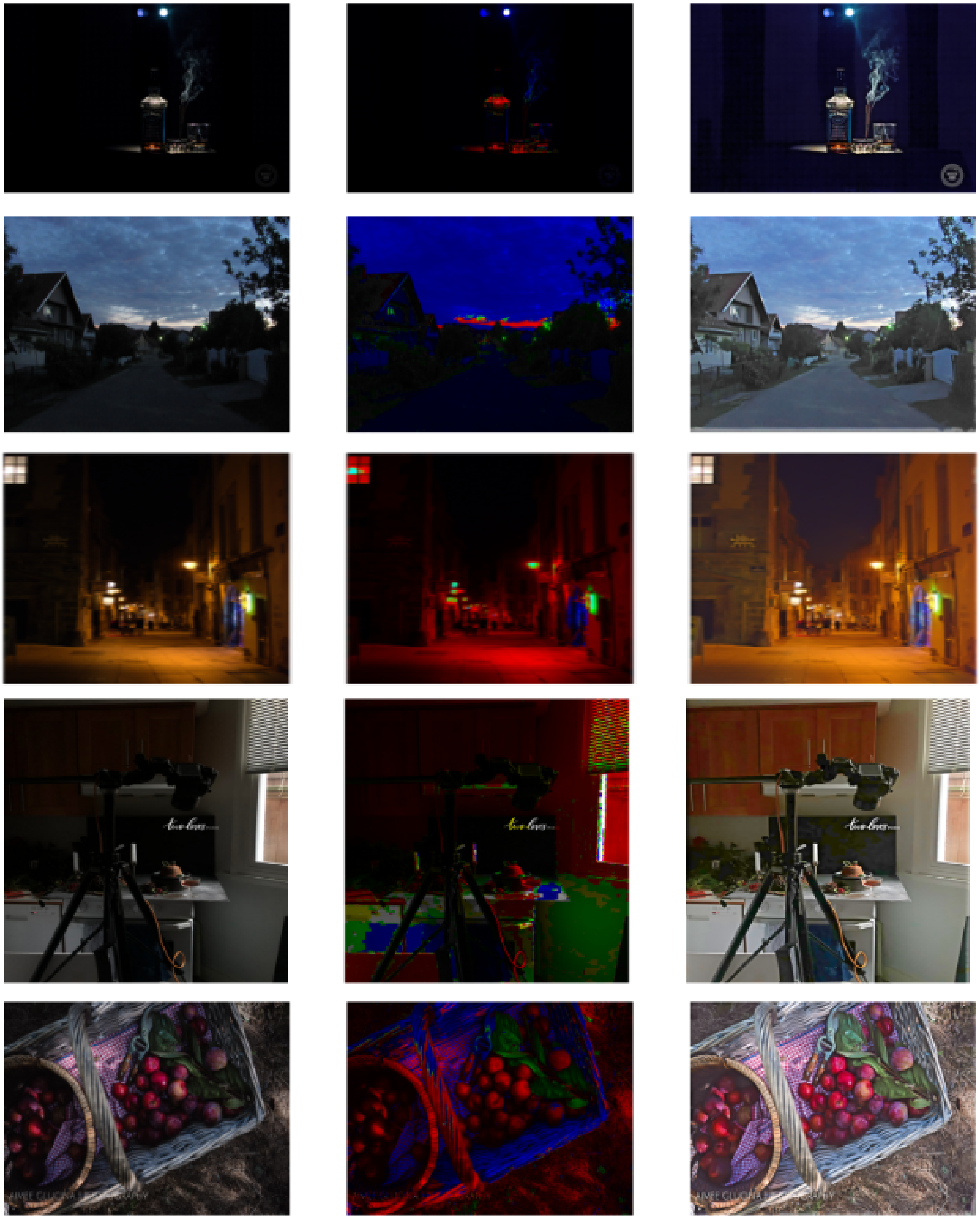

Fig. 1. **First Column :** Natural low-light images. **Middle Column:** The illumination maps estimated by our method. **Right Column:** The results enhanced by our method.

with zero. An initial estimate of the illumination map $\hat{\mathbf{T}}$ for each pixel p, is obtained as per the following operation:

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

This initial estimation of illumination map enables to handle non-uniform illuminations and is based on principle that the illumination is at least the maximal value of three channels at a certain location. $\hat{\mathbf{T}}$ also ensures that pixels of recovery image $\mathbf{R}$ do not saturate.

To preserve the overall structure and smooth the textural details with the initial illumination map $\hat{\mathbf{T}}$ following optimization problem is proposed:

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

where α is the coefficient to balance the involved two terms and, $\|\cdot\|_F$ and $\|\cdot\|_1$ designate the Frobenious and l_1 norms, respectively. Further, $\mathbf{W}$ is the weight matrix, and $\nabla \mathbf{T}$ is the first order derivative filter.

A. Solving the Optimization Problem

The objective function (4) involve $\mathbf{T}$ in both the terms an auxiliary variable $\mathbf{G}$ is replaced with $\nabla \mathbf{T}$ for making the problem separable and thus easy to solve. Accordingly $\nabla \mathbf{T} = G$ is added as a constraint. As a result, the earlier optimization problem is equivalent to the following:

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

The augmented Lagrangian function (5) for this problem can be formulated as:

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

$\Phi(\mathbf{Z}, \nabla \mathbf{T} - \mathbf{G}) = \frac{\mu}{2} \|\nabla \mathbf{T} - \mathbf{G}\|_F^2 + \langle \mathbf{Z}, \nabla \mathbf{T} - \mathbf{G} \rangle$, 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 $\mathbf{T}$ from (6) gives the problem as following:

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

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

$$\begin{gathered} 2(\mathbf{T} - \hat{\mathbf{T}}) + \mu^{(k)} \mathbf{D}^T \left(\mathbf{D}\mathbf{T} - \mathbf{G}^{(k)}\right) + \mathbf{D}^T \mathbf{Z}^{(k)} = 0 \\ \Rightarrow \left(2\mathbf{I} + \mu^{(k)} \mathbf{D}^T \mathbf{D}\right) \mathbf{T} = 2\hat{\mathbf{T}} + \mu^{(k)} \mathbf{D}^T \left(\mathbf{G}^{(k)} - \frac{\mathbf{Z}^{(k)}}{\mu^{(k)}}\right) \end{gathered} \tag{8}$$

where I denotes the identity matrix and $(\cdot)^T$ is the transpose operator. The operations $\mathbf{DX}$ and $\mathbf{D^T V}$ represent reshape $(\mathbf{Dx})$ and reshape $\left(\mathbf{D^T v}\right)$, respectively, where $\mathbf{X} \in R^{M \times N}, \mathbf{x} = \operatorname{vec}(\mathbf{X}), \mathbf{V} \in R^{2M \times N}$ or $R^{M \times 2N}, \mathbf{v} = \operatorname{vec}(\mathbf{V})$. The operator $\operatorname{vec}(\cdot)$ vectorizes a matrix and 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. D represents the finite differences approximation of the horizontal and vertical gradients [13]. Thus, $\mathbf{D} = \left[\begin{array}{cc} \mathbf{D}_x^T & \mathbf{D}_y^T \end{array}\right]^T$, such that:

$$\begin{aligned} \mathbf{Dx} &= \left[\begin{array}{c} \mathbf{D}_x \\ \mathbf{D}_y \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathbf{D}_x \mathbf{x} \\ \mathbf{D}_y \mathbf{x} \end{array}\right] \\ \mathbf{D}^T \mathbf{v} &= \left[\begin{array}{cc} \mathbf{D}_x^T & \mathbf{D}_y^T \end{array}\right] \left[\begin{array}{c} \mathbf{v}_a \\ \mathbf{v}_b \end{array}\right] = \mathbf{D}_x^T \mathbf{v}_a + \mathbf{D}_y^T \mathbf{v}_b \end{aligned} \tag{9}$$

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:

$$\mathbf{D}_x \mathbf{x} = \operatorname{vec}\left(d_x * \mathbf{X}\right), \mathbf{D}_y \mathbf{x} = \operatorname{vec}\left(d_y * \mathbf{X}\right) \tag{10}$$

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

$$d_x = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{array}\right], \quad d_y = \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{array}\right] \tag{11}$$

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

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

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:

$$\begin{aligned} \mathbf{D}_x &= \left[\begin{array}{cccccc} -1 & 0 & \dots & 1 & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & \ddots & 0 & 1 \\ 1 & \ddots & \ddots & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & 0 & 1 & 0 & \dots & -1 \end{array}\right] \\ \mathbf{D}_y &= \left[\begin{array}{ccccc} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \\ 1 & 0 & \dots & 0 & -1 \end{array}\right] \end{aligned} \tag{13}$$

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:

$$\mathbf{T}^{(k+1)} \leftarrow \mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(2\hat{\mathbf{T}} + \mu^{(k)}\mathbf{D}^T\left(\mathbf{G}^{(k)} - \frac{\mathbf{Z}^{(k)}}{\mu^{(k)}}\right)\right)}{\mathbf{2} + \mu^{(k)}\sum_{i\in\{x,y\}}\overline{\mathcal{F}\left(d_i^e\right)} \circ \mathcal{F}\left(d_i^e\right)}\right) \tag{14}$$

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 **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:

$$\mathbf{D}^T\mathbf{D} = \mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_y^T\mathbf{D}_y \tag{15}$$

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,

$$\begin{aligned} &d_i^f * d_i = d_i \oplus d_i, i \in \{x, y\} \\ \Rightarrow &\mathcal{F}\left(d_i \oplus d_i\right) = \overline{\mathcal{F}\left(d_i\right)} \circ \mathcal{F}\left(d_i\right) \end{aligned} \tag{16}$$

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:

$$\begin{aligned} \mathbf{DX} &\equiv \text{reshape}\left(\begin{bmatrix} \text{vec}\left(\mathbf{X}_H\mathbf{D}_{H1}\right) \\ \text{vec}\left(\mathbf{D}_{V1}\mathbf{X}_V\right) \end{bmatrix}\right) \\ \mathbf{D}^T\mathbf{V} &\equiv \mathbf{V}_H\mathbf{D}_{H2} + \mathbf{D}_{V2}\mathbf{V}_V \end{aligned} \tag{17}$$

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

$$\begin{aligned}
\mathbf{D}_{V1} &= \begin{bmatrix} -1 & 1 & 0 & \dots & \dots \\ 0 & -1 & 1 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix} \\
\mathbf{D}_{H2} &= \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \\ 1 & 0 & \dots & 0 & -1 \end{bmatrix} \\
\mathbf{X}_H &= \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} & x_{11} \\ x_{21} & x_{22} & \dots & x_{2N} & x_{21} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MN} & x_{M1} \end{bmatrix} \\
\mathbf{X}_V &= \begin{bmatrix} x_{11} & x_{12} & \dots & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ x_{M1} & x_{M2} & \dots & \dots & x_{MN} \\ x_{12} & x_{13} & \dots & x_{1N} & x_{11} \end{bmatrix} \\
\mathbf{V}_V &= \begin{bmatrix} v_{bMN} & v_{bM1} & v_{bM2} & \dots & v_{bMN-1} \\ v_{b11} & v_{b12} & \dots & \dots & v_{b1N} \\ v_{b21} & v_{b22} & \dots & \dots & v_{b2N} \\ \vdots & & \ddots & \ddots & \vdots \\ v_{bM1} & v_{bM2} & \dots & \dots & v_{bMN} \end{bmatrix}
\end{aligned} \tag{18}$$

where x_{ij} are the elements of $\mathbf{X}$ and v_{bij} are the elements of $\mathbf{V}_b$. Thus, (14) can be implemented using (13) and the matrices listed above.

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

$$\mathbf{G}^{(k+1)} \leftarrow \arg\min_{\mathbf{G}} \|\mathbf{W} \circ \mathbf{G}\|_1 + \Phi\left(\mathbf{Z}^{(k)}, \nabla\mathbf{T}^{(k+1)} - \mathbf{G}\right) \tag{19}$$

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

$$\mathbf{G}^{(k+1)} = \mathcal{S}_{\frac{\alpha\mathbf{W}}{\mu^{(k)}}}\left[\nabla\mathbf{T}^{(k+1)} + \frac{\mathbf{Z}^{(k)}}{\mu^{(k)}}\right] \tag{20}$$

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

$$\mathcal{S}_e[a] = \text{sgn}(a)\max(|a| - e, 0) \tag{21}$$

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 :

$$\begin{aligned} &\mathbf{Z}^{(k+1)} \leftarrow \mathbf{Z}^{(k)} + \mu^{(k)} \left(\nabla \mathbf{T}^{(k+1)} - \mathbf{G}^{(k+1)} \right) \\ &\mu^{(k+1)} \leftarrow \mu^{(k)} \rho, \rho > 1 \end{aligned} \tag{22}$$

B. Weighting Strategies

For the structure-aware refinement on the initial illumination map, three possible weighing strategies can be used for the design of $\mathbf{W}$, such that $\mathbf{W} = \mathrm{reshape}(\mathbf{w})$, where $\mathbf{w} = \left[\begin{array}{cc} \mathbf{W}_x^T & \mathbf{W}_y^T \end{array} \right]^T$.

Strategy I: It can be seen that setting the weight matrix as

$$\mathbf{W}_x(p) \leftarrow 1, \quad \mathbf{W}_y(p) \leftarrow 1 \tag{23}$$

leads (4) to a classic loss total variation minimization problem.

Strategy II: use the gradient of the initial illumination map as the weight

$$\mathbf{W}_x(p) \leftarrow \frac{1}{|\nabla_x \hat{\mathbf{T}}(p)| + \epsilon}, \mathbf{W}_y(p) \leftarrow \frac{1}{|\nabla_y \hat{\mathbf{T}}(p)| + \epsilon} \tag{24}$$

Strategy III: This involves the use of relative total variation, so for each location, the weight is set as:

$$\begin{aligned} \mathbf{W}_x(p) &\leftarrow \sum_{q \in \Omega(p)} \frac{G_\sigma(p,q)}{\left| \sum_{q \in \Omega(p)} G_\sigma(p,q) \nabla_x \hat{\mathbf{T}}(q) \right| + \epsilon} \\ \mathbf{W}_y(p) &\leftarrow \sum_{q \in \Omega(p)} \frac{G_\sigma(p,q)}{\left| \sum_{q \in \Omega(p)} G_\sigma(p,q) \nabla_y \hat{\mathbf{T}}(q) \right| + \epsilon} \end{aligned} \tag{25}$$

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

$$\nabla \mathbf{X} = \left[\begin{array}{c} \nabla_x \mathbf{X} \\ \nabla_y \mathbf{X} \end{array} \right] \equiv \left[\begin{array}{c} \mathbf{D}_x \mathbf{x} \\ \mathbf{D}_y \mathbf{x} \end{array} \right] \tag{26}$$

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

$$G_\sigma(p,q) \propto \exp\left(-\frac{\|\mathbf{p} - \mathbf{q}\|^2}{2\sigma^2} \right) \tag{27}$$

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:

$$\mathbf{T} \leftarrow \mathbf{T}^\gamma, \gamma < 1 \tag{28}$$

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. Block-matching and 3-D filtering (BM3D)[14] is proposed

In BM3D 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 3D 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 weightaveraged before replacement. BM3D is used as an off-theshelf 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:

$$\mathbf{R}_f \leftarrow \mathbf{R} \circ \mathbf{T} + \mathbf{R}_d \circ (\mathbf{1} - \mathbf{T}) \tag{29}$$

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

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
              $\alpha$, $\gamma$, $\rho$,and $\mu^(0)$
  Initialize: $k = 0$, $\mathbf{G}^{(0)} = \mathbf{Z}^0 = \mathbf{0} \in R^{2M \times N}$
  Estimate the initial illumination map $\hat{\mathbf{T}}$ on $\mathbf{L}$ using (3)
  while $k < k_o$ do
      Update $\mathbf{T}^{(k+1)}$ $using$ (14);
      Update $\mathbf{G}^{(k+1)}$ $using$ (20);
      Update $\mathbf{Z}^{(k+1)}$ $and$ $\mu^{(k+1)}$ using (22);
      $k = k + 1$;
  end
  Apply Gamma correction on $\mathbf{T}^{(k_o)}$ $using$(28);
  Obtain $\mathbf{R}$ using Gamma corrected $\mathbf{T}^{(k_o)}$ $using$(2);
  Denoise the result using BM3D using (29);
  Output    : final enhanced image
```

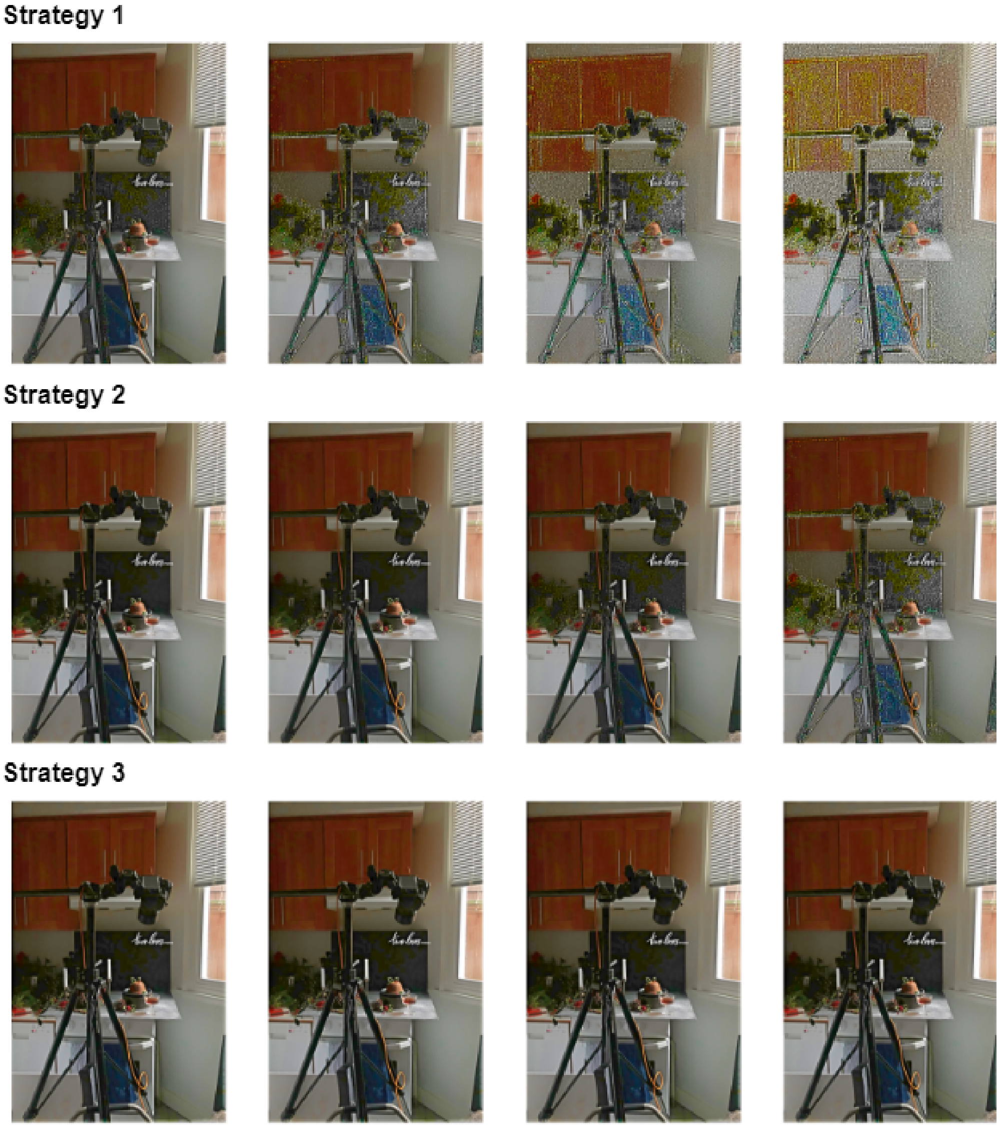

Fig. 2. Difference between the weighting strategy I and II with varying parameter α. For the first row (Strategy I), we use $\alpha \; \epsilon \; \{0.2,\ 0.5,\ 1.0,\ 3.0\}$ corresponding to the four results, respectively. While for the second row (Strategy II), we use $\alpha \; \epsilon \; 0.15 \times \{0.2,\ 0.5,\ 1.0,\ 3.0\}$ to do the test. The choice of the coefficient 0.15 is based on the observation that the difference norms (DN), defined as $||\mathbf{T} - \hat{}\mathbf{T}||F$, of the two cases are close, so that the comparison is fair.

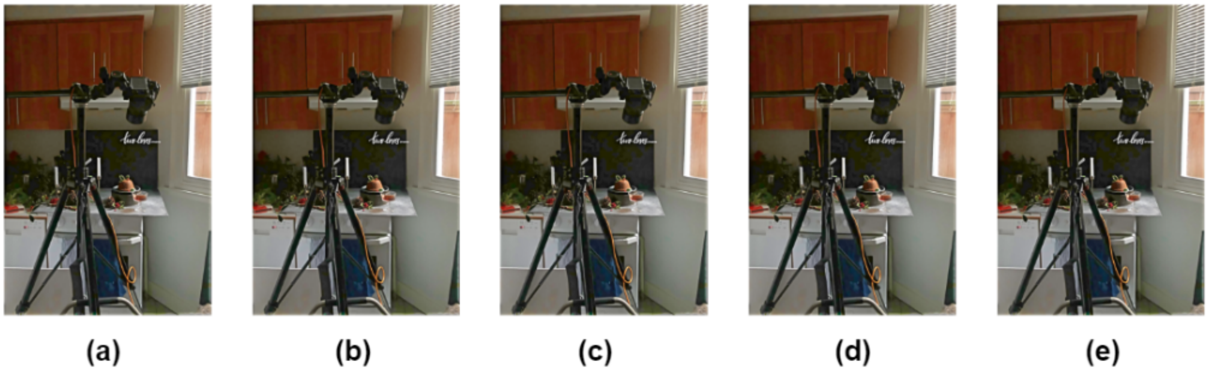

Fig. 3. Effect of parameter σ for weighting strategy III. **From (a) to (e):** the results by setting σ to $10^{(-5)}$, 1, 2, 3 and 4 respectively with fixed $\alpha = 0.15 \times 0.2$

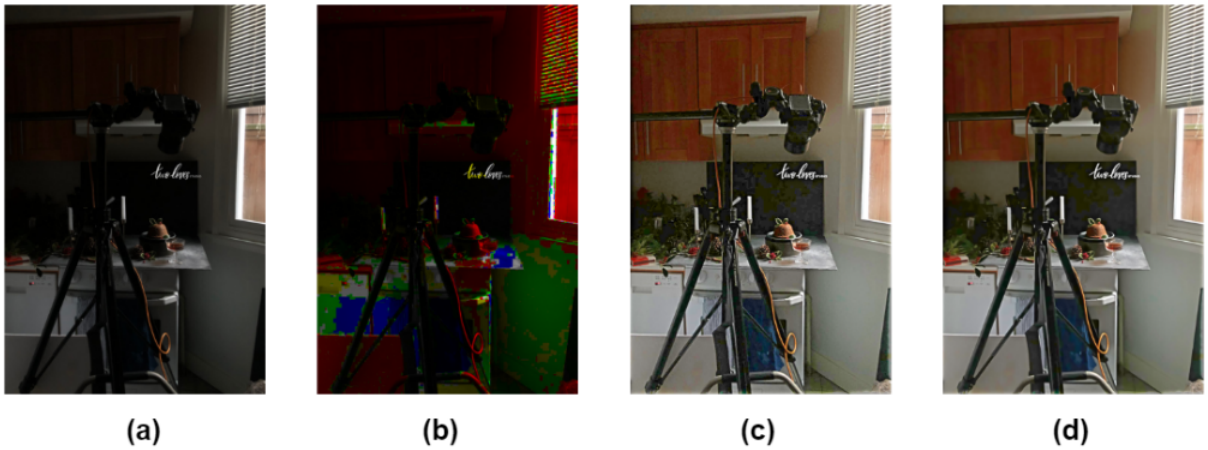

Fig. 4. Comparison of results without and with denoising. (a) Input . (b) Illumination map. (c) Without denoising. (d) With denoising

3. CONCLUSION

We have implemented an efficient and successful way for enhancing low-light photographs in this project. We observe that the accuracy with which the illumination map is computed, is critical to low-light enhancement. To increase lighting uniformity, the structure-aware smoothing model was created. We implemented that only a part of the paper in which an algorithm is proposed that solves the approximation issue in a substantial amount of time. Furthermore, this strategy is applicable to other weighting schemes as well. It is promising that, this low-light picture enhancement approach can provide high visibility inputs to numerous vision-based applications, such as edge detection, feature matching, object recognition, and tracking, and therefore increase their performance.

Acknowledgment

We would like to thank Prof. Debdoot Sheet for teaching the course Digital Image Processing and for evaluating this term project. It was a great learning experience for us. Finally we would like to thank the developers and maintainers of Numpy [15], OpenCV [16] and Scikit-Image [17] repositories, without which most of our work would have been impossible.

4. REFERENCES

[1] Etta D. Pisano, Shuquan Zong, Bradley M. Hemminger, Marla DeLuca, R. Eugene Johnston, Keith Muller, M. Patricia Braeuning, and Stephen 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, pp. 193–200, Nov. 1998.

[2] H.D. Cheng and X.J. Shi, "A simple and effective histogram equalization approach to image enhancement," *Digital Signal Processing*, vol. 14, no. 2, pp. 158–170, Mar. 2004.

[3] M. Abdullah-Al-Wadud, Md. Hasanul Kabir, M. Ali Akber Dewan, and Oksam Chae, "A dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 593–600, 2007.

[4] Edwin Herbert 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, 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, 1997.

[7] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li, "Naturalness preserved enhancement algorithm for non-uniform illumination images," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3538–3548, 2013.

[8] Xueyang Fu, Delu Zeng, Yue Huang, Yinghao Liao, Xinghao Ding, and John Paisley, "A fusion-based enhancing method for weakly illuminated images," *Signal Processing*, vol. 129, pp. 82–96, 2016.

[9] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding, "A weighted variational model for simultaneous reflectance and illumination estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2782–2790.

[10] Xuan Dong, Guan Wang, Yi Pang, Weixin Li, Jiangtao Wen, Wei Meng, and Yao Lu, "Fast efficient algorithm for enhancement of low lighting video," in *2011 IEEE International Conference on Multimedia and Expo*, 2011, pp. 1–6.

[11] Xiaojie Guo, Yu Li, and Haibin Ling, "Lime: Low-light image enhancement via illumination map estimation," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 982–993, 2017.

[12] Jun Xu, Yingkun Hou, Dongwei Ren, Li Liu, Fan Zhu, Mengyang Yu, Haoqian Wang, and Ling Shao, "STAR: A structure and texture aware retinex model," *IEEE Transactions on Image Processing*, vol. 29, pp. 5022–5037, 2020.

[13] Gordon Wetzstein, "EE 367 / CS 448i computational imaging and display notes: Image deconvolution (lecture 6)," p. 8.

[14] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[15] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi,

Christoph Gohlke, and Travis E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sept. 2020.

[16] Gary Bradski and Adrian Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*, ” O’Reilly Media, Inc.”, 2008.

[17] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors, “scikit-image: image processing in Python,” *PeerJ*, vol. 2, pp. e453, 6 2014.