UM-Based Image Enhancement in Low-Light Situations

SHWU-HUEY YEN*  CHUN-HSIEN LIN  HWEI-JEN LIN  JUI-CHEN CHIEN
Department of Computer Science and Information Engineering
Tamkang University,
151 Ying-chuan Road, Tamsui, New Taipei City, Taiwan 25137
REPUBLIC OF CHINA (ROC)
e-mail*: 105390@mail.tku.edu.tw

Abstract: - Unsharp masking (UM) is an effective and popular method on image enhancement. However, it is sensitive to noise and tends to have over/under shooting problems. In this paper, we propose an improved UM-based technology for image enhancement. First, noises are detected and smoothed. Then, integrating the silhouette and crease edges (major and minor edges), we design an adaptive weighting method to enhance the contrast for edges. In this way, the major edges (silhouette) are sharpened more comparing to minor edges (crease). Hence, not only the over/under shooting problems are solved but the contrast on edges are properly enhanced. The proposed method has been compared to existing UM-based methods and the results are satisfying.

Key-Words: - Unsharp Masking, Canny Edge Detector, Contrast Enhancement, Laplacian Filter, Connected Component Analysis.

1 Introduction
When taking digital photos in low light conditions, CCD (Charge-coupled Device) or CMOS (Complementary Metal Oxide Semiconductor) sensor chips take longer exposure time to capture more light. However, longer exposure time will cause camera shake if a tripod is not used. A solution commonly adopted by automatic digital camera is to use a higher ISO. ISO means the amount of sensitivity of light falling on sensor. The higher the ISO, the more sensitive the image sensor and therefore the possibility to take pictures in low-light situations. However, a higher ISO also causes noises. The noise occurs because the physical properties of light-sensitive components, for example, read noise, dark current noise, fixed pattern noise, etc. These noises greatly reduce the image quality. How to sharpen and de-noise images taken in low light conditions is concerned in this study.

Image sharpening is an important issue to many subsequent images processing tasks or simply for visual quality of images. There are many methods for image sharpening, for example, adaptive histogram equalization (AHE) and adaptive contrast enhancement (ACE). Comparing with these methods, Unsharp Masking (UM) has been proposed and has a good image enhancement effect. Using a high-pass filter, UM methods enhance an image by adding back a scaled high frequency as in the Eq(1).

\[ Y(n,m) = I(n,m) + \lambda \times Z(n,m), \quad (1) \]

where \(I(n, m)\) is the original image, \(Z(n, m)\) is the high-frequency portion of the original image produced by a high-pass filter, \(\lambda\) is a (global) scaling factor, and \(Y(n, m)\) is the enhanced image.

There are many ways for generating \(Z(n, m)\). One of the most basic linear methods is Laplacian filter, but there are two drawbacks. It increases the sensitivity of the noise. And it may cause over shooting problem in the high-frequency part and under shooting problem in the low-frequency part of the enhanced image.

To effectively suppress noises, non-linear polynomial operator is often used as in [1]-[3]. Cubic Unsharp Masking (CUM) is one of the most representative method [1]. CUM effectively suppresses noises when the image is moderately damaged, but it enhances the noises when the image is seriously damaged. In addition, it tends to have the over/under shooting along the borders of edges. To solve the over/under shooting problem, adaptive UM [2] was proposed such that they used different scaling factors for high-, medium-, and low-frequency parts of the image. But the price is the complicated algorithm with many parameters. It may not be easy to choose suitable parameters for a given image. In [3], improved from the method of [1], it used a cascaded configuration of cubic unsharp masking (CS-CUM) to simultaneously
remove image noises and improve image quality. Authors especially emphasized on the continuity of the “edge”. According to [3], the proposed method can effectively reduce the noise amplification value about one-third along edges comparing to that in [1]. However, in smooth area, not only it fails to suppress the noises but also it amplifies the noise. Kim & Cho probed the relationship between textures and noises in [4]. They classified four kinds of textures that they all share the same property of large local variances just like noises do. The proposed method helps to clear out texture and noise somehow, but it did not suppress noises and can not distinguish all possible textures from noises.

Recently, the research in image rendering of computer graphics has a progressive development due to the popularity of video games, simulators, movie or TV special effects, etc. To have a better visual effect, the contours of interested objects are usually enhanced. Particularly, edges are classified as silhouettes and creases according to whether they are major edges or minor edges. A silhouette will be enhanced more and a crease will only be enhanced moderately [6]-[9]. In this paper, we adopt the concept of silhouette and crease to adaptively enhance a low-light noisy image. The proposed method is UM-based which will first identify and smooth noises, then classify edges into silhouettes and creases. In this way, proper scaling weights can be assigned and, therefore, edges are enhanced appropriately. The outline of the proposed algorithm is given in Fig. 1.

2 Proposed Method

For a given grayscale image \(I\), it is used to generate a de-noised image \(D\) and a high-frequency image \(L\). Canny operator is then applied on \(D\) to detect edges. Silhouette and crease are classified. Adaptive weights are assigned to silhouettes and creases by Gaussian smoothing. By multiplying the weighting matrix \(W\) and \(L\) position-wise, the high frequency part of \(I\) consists of true edges only and more appropriately represented. Finally, the enhanced image is obtained as in Eq. (2).

\[
O = D + \lambda \times (W \times L)
\]

where \(O\) is the output image, \(\lambda\) is the scaling factor, \(L\) is the image after implementation of the Laplacian filter on \(I\), \(W\) is the weighting matrix (its size is the same as the image \(I\)’s), and \(D\) is the image after noise removal. The details are given in the following.

2.1 Noise Detection and Suppression

We use Laplacian operator to detect pixels where intensity changes. Fig. 2 is a night view of Tamsui River (in New Taipei City, Taiwan), (a) is the original image \(I\), (b) is the Laplacian filtered image \(L\). The magnitude of \(L\) of a point reflects how large the intensity changes at this point. We use a threshold, \(th_{\text{Noise}}\), to determine whether a point is a candidate for noise. A point on \((x, y)\) is a noise candidate if \(|L(x,y)| \geq th_{\text{Noise}}\).

![Fig. 1 Flow diagram of the proposed algorithm](image-url)
Noise candidates include both noises and edge points. By observing many images taken in low light conditions, the noises from high ISO have blobs with size not larger than $3 \times 3$, whereas the edges points usually form blobs larger than that. Therefore, by connected component (CC) analysis on those noise candidates, we label those candidates to be noises if the CC has the size not larger than $3 \times 3$. To smooth noises, the intensity of every noise point $P$ is replaced by the average intensity value taken from non-noise points of a $5 \times 5$ window on $I$ (centered at $P$). Fig. 2(c) shows the smoothed result $D$.

### 2.2 Edge Detection and Classification

Canny edge detector is applied on the smoothed image $D$ to find edges. Canny edge detector has three parameters, Gaussian blur of $\sigma$, double thresholds $T_1$ & $T_2$ ($T_2 > T_1$). A larger $\sigma$ is more suitable when the image noise is severe. Fig. 3(a) is the result of edge detection from Fig. 2(c) with $\sigma = 0.6$, $T_1 = 150$ & $T_2 = 200$.

Improving the contrast of contour and details is critical for an image to have a good visual quality. However, the principal contours deserve more enhancement than fine details do. Therefore, we adopt the concept of silhouette and crease, and assign them different enhancement weights. To distinguish between silhouettes, principal contours or major edges, and creases, fine details or minor edges, we take the lengths of curve into consideration. Taking two thresholds, $L_1$ and $L_2$ ($L_1 < L_2$), a continuous curve is a silhouette if its length is at least $L_2$, and a crease if its length is between $L_1$ and $L_2$. Finally, the rest of edges are eliminated since they are not important and mostly are noises. Fig. 3(b) illustrates the classification of silhouette and crease. Comparing (a) and (b), noises are eliminated further in (b).

![Fig. 2 The night view of Tamsui River](image)

![Fig. 3 Edge detection and classification of $D$ on Fig.2(c)](image)
2.3 The Assignment of Adaptive Weights

To assign weights to silhouette/crease, two binary images are created. One is having ones on those points belonging to silhouettes (image S) and the other is having ones on those points belong to creases (image C). To S, a morphological dilation with a structuring element of $3 \times 3$ is applied and followed by a Gaussian blurring. To C, a Gaussian blurring is applied. Now combine these two images into a weighting matrix $W$ as in Eq.(3).

$$
\begin{align*}
    w(i,j) = \begin{cases} 
        s(i,j) & \text{if } c(i,j) = 0 \text{ and } s(i,j) \neq 0, \\
        c(i,j) & \text{if } c(i,j) \neq 0 \text{ and } s(i,j) = 0, \\
        \max(s(i,j),c(i,j)) & \text{if } c(i,j) \neq 0 \text{ and } s(i,j) \neq 0, \\
        0 & \text{otherwise.}
    \end{cases}
\end{align*}
$$

where $w(i,j)$ is the weight assigned to a point on $(i,j)$ position, $s(i,j)$ and $c(i,j)$ are the values on point $(i,j)$ in images $S$ and $C$ respectively.

2.4 Image Enhancement

Finally, UM technology for image enhancement is implemented as in Eq. (2). Comparing to the traditional UM method, our method smoothes noises first to avoid noises erroneously magnified, and soothes the over/under shooting problem and gives a natural look on edges on the enhanced image since weights are smoothed by Gaussian blurring. Fig. 4 shows the final enhancement result.

![Fig. 4 The final enhanced result of Fig. 2(a)](image)

3 Experimental results and analysis

Our method has several parameters: the enhancement scaling factor $\lambda$ in Eq. (2), $th_{noise}$ in determining the noise candidates, $\sigma$, $T_1$, $T_2$ in Canny edge detector, $L_1$, $L_2$ in determining silhouette and crease. Table 1 gives the experimental values that they give satisfactory results in general.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$th_{noise}$</th>
<th>$\sigma$</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$L_1$</th>
<th>$L_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.35</td>
<td>25</td>
<td>0.6</td>
<td>150</td>
<td>200</td>
<td>20</td>
<td>70</td>
</tr>
</tbody>
</table>

Although the parameters setting in Table 1 is applicable to images in general, however, it may not be suitable when the image is severely damaged. For example, when Gaussian noise of variance 50 is added, there are many noise candidates and it is possible that noises also form large blobs. Then these noises will be mistaken to be edge points and magnified in later process. Thus, for simplification, in a fixed $T_1$, $T_2$, $L_1$, $L_2$ consideration, we take any small rectangular area from smooth background to serve as an estimate of the noises. Let $v$ be the intensity variance of the selected area. Then parameters $\sigma$ and $th_{noise}$ are adjusted accordingly as in Eq. (4) & (5). Observe that, in Eq.(4), logarithm is taken in $v$, so small variations on $v$ do not affect the results.

$$
\begin{align*}
    \sigma &= \begin{cases} 
        0.6 & \text{if } v \leq 60, \\
        0.5231 \times \log_{10} v - 0.3416 & \text{if } v > 60,
    \end{cases} \\
    th &= \begin{cases} 
        15 & \text{if } v \leq 60, \\
        25 & \text{if } v \geq 300, \\
        30.385 \times \sigma - 3.4231 & \text{if } 60 < v \leq 300.
    \end{cases}
\end{align*}
$$

As for the scaling factor $\lambda$, it gives good results for values between 0.35 and 0.5 (0.35 is used throughout the tests).

Three existing methods are compared: Cubic Unsharp Masking (CUM) [1], Cascade of CUM (CS-CUM) [3], and Kim’s Feature and Noise Adaptive UM (Kim’s) [4]. Parameter settings are adopted from their papers. The tested image is taken from Tamsui River with Olympus E-510 and ISO 800. Fig. 5 shows the enhanced results by different methods. Noises are erroneously magnified in all (b)–(d). We further examine the enhanced results in two areas, smooth and textured as indicated in Fig. 5(a). The enlarged corresponding areas and the variances are shown in Table 2. As we know, after enhancement, the smaller variance is better in smooth area, and the larger variance is the better in texture area. The figures in Table 2 do confirm that the proposed method effectively suppress the noise (with variance 13.56 even smaller than the original’s) and properly enhanced edges (with variance 4779.59 larger than the original’s).
4 Conclusion
This paper presented a UM-based method to sharpen and de-noise low-light images. Rules of choosing parameters for the algorithm are provided. Unlike traditional UM, our method first detects and suppresses noises. Also, to have a more natural enhancement visual effect along edges, silhouette and crease edges are classified and different weights are assigned. By this way, the problem of noises caused by high ISO for images taken in low light conditions is solved. The method outperformed existing methods.

![Image of enhanced results](image)

Fig. 5 The enhanced results of different methods where two yellow boxes in (a) indicated the areas for later comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Original</th>
<th>Kim’s</th>
<th>CUM</th>
<th>CS-CUM</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth area variance</td>
<td>22.57</td>
<td>60.91</td>
<td>68.51</td>
<td>40.11</td>
<td>13.56</td>
</tr>
<tr>
<td>Texture area variance</td>
<td>4406.97</td>
<td>5018.87</td>
<td>5019.01</td>
<td>4889.52</td>
<td>4779.59</td>
</tr>
</tbody>
</table>
References:


