Article

Detail Enhancement Multi-Exposure Image Fusion Based on Homomorphic Filtering

Yunxue Hu 1,2, Chao Xu 1,2,*, Zhengping Li 1,2, Fang Lei 3, Bo Feng 1,2, Lingling Chu 1,2, Chao Nie 1,2 and Dou Wang 1,2

1 School of Integrated Circuits, Anhui University, Hefei 230601, China; p20201024@stu.ahu.edu.cn (Y.H.); 041738@ahu.edu.cn (Z.L.); p18101008@stu.ahu.edu.cn (B.F.); p21101008@stu.ahu.edu.cn (L.C.); p19301109@stu.ahu.edu.cn (C.N.); p20301228@stu.ahu.edu.cn (D.W.)
2 AnHui Engineering Laboratory of Agro-Ecological Big Data, Hefei 230601, China
3 School of Humanities, Shanghai University of Finance and Economics, Shanghai 200433, China; leifang296@163.com
* Correspondence: xchao@ahu.edu.cn; Tel.: +86-13339199368

Abstract: Due to the large dynamic range of real scenes, it is difficult for images taken by ordinary devices to represent high-quality real scenes. To obtain high-quality images, the exposure fusion of multiple exposure images of the same scene is required. The fusion of multiple images results in the loss of edge detail in areas with large exposure differences. Aiming at this problem, this paper proposes a new method for the fusion of multi-exposure images with detail enhancement based on homomorphic filtering. First, a fusion weight map is constructed using exposure and local contrast. The exposure weight map is calculated by threshold segmentation and an adaptively adjustable Gaussian curve. The algorithm can assign appropriate exposure weights to well-exposed areas so that the fused image retains more details. Then, the weight map is denoised using fast-guided filtering. Finally, a fusion method for the detail enhancement of Laplacian pyramids with homomorphic filtering is proposed to enhance the edge information lost by Laplacian pyramid fusion. The experimental results show that the method can generate high-quality images with clear edges and details as well as similar color appearance to real scenes and can outperform existing algorithms in both subjective and objective evaluations.

Keywords: dynamic range; exposure fusion; homomorphic filtering; gaussian curve; laplacian pyramid

1. Introduction

In natural scenes, the dynamic range can reach six orders of magnitude, and the normal human eye can capture four orders of magnitude, dynamic range refers to the ratio between the maximum brightness and the minimum brightness in the same scene, while the dynamic range that can be captured by ordinary digital cameras [1] is only two orders of magnitude [2]; therefore, the images captured by ordinary digital cameras are exposed and poor areas can cause a loss of detail. High Dynamic Range (HDR) images can truly reflect natural scenes [3].

The generation of HDR images is divided into two methods, hardware and software. HDR equipment can directly obtain images of natural scenes; however, dedicated equipment is expensive, and HDR images cannot be displayed on ordinary low dynamic range (LDR) equipment, and thus it is difficult to use [4]. The software method uses HDR imaging technology to make the images displayed by LDR devices richer in detail. The technique includes two methods, tone mapping [5,6], and multi-exposure image fusion. The first requires estimating the camera response function (CRF) to construct an HDR image and then uses tone mapping to display the HDR image on a normal LDR display device; however, this has a disadvantage.
The calculation of CRF requires multiple exposure parameters, and these exposure parameters need to be calculated separately. Thus, it is time-consuming and limited [7]. In contrast, the computationally simple multi-exposure image fusion method is more efficient as it directly fuses multiple images of different exposures of an HDR scene into a high-quality image that can be displayed on an LDR device [8].

Existing traditional multi-exposure fusion methods can be divided into two categories: based on the spatial domain and based on the transform domain. The spatial domain mainly analyzes and operates on pixel values or structural blocks. Its three-dimensional information is rich [9], the method is simple, and the calculation cost is low; however, the generated images will have problems, such as noise and information loss. For example, Gu et al. [10] used the structural tensor fusion of the input image to iteratively correct the gradient field using quadratic mean filtering and multi-scale nonlinear compression; however, this method made the image noisy.

Li et al. [11] used local contrast, brightness, and color dissimilarity to construct a weight map, followed by recursive filtering for denoising and refinement, and finally, weighted fusion to obtain the result; however, this method produced unnatural artifacts. Huang et al. [12] decomposed the image into contrast extraction, structure preservation, and intensity adjustment, where structure preservation and intensity adjustment were calculated by local weights, global weights, and saliency weights, and finally reconstructed the resulting map. The details of the images generated by this method were well preserved, but the colors were quite different from the input images.

In the transform domain, the image is transformed into the frequency domain through discrete Fourier transform (DFT) [13], pyramid transform [14], etc., and the image color generated by these methods is very close to reality; however, it is easy to lose texture details [15]. For example, Mertens et al. [16] used contrast, saturation, and good exposure to construct a weight map, and finally used multi-resolution fusion, which can generate images with natural colors but cannot preserve the full edge texture details of the image.

Wang et al. [17] added local Laplacian filtering to the Mertens method to enhance details in local overexposed and underexposed regions and proposed discrete sampling and interpolation to speed up the results. With the development of machine learning, methods for multi-exposure image fusion with neural networks have begun to appear. Xu et al. [18] proposed a multi-exposure image fusion method based on generative adversarial networks, in which the generator network and the discriminator network are simultaneously trained to form an adversarial relationship, and a self-attention mechanism was introduced for the problem of large image exposure differences. In addition, some new fusion methods have gradually emerged.

For example, Yang et al. [19] first used the K-means-based algorithm (K-SVD) to calculate the sparsity exposure dictionary (SED) to construct the exposure estimation map and then used the exposure estimation map and adaptive guided filter to construct the final fusion decision map and finally pyramid fusion. The method of Ulucan et al. [20] utilizes the histogram and K-means classified atlas to extract linear embedding weights and watershed masks for fusion and finally corrects unsatisfactory color intensities.

In this paper, the transform domain pyramid fusion method is used and based on the Mertens method, threshold segmentation and an adaptively adjustable Gaussian curve are proposed to calculate the exposure weight, and the homomorphic filter is used to enhance the detail layer of the Laplacian pyramid. This method not only generates images with rich colors but also preserves the texture details of the images. The image fusion algorithm proposed in this paper has three main contributions:

1. This paper applies homomorphic filtering to the multi-exposure image fusion algorithm for the first time. Other detail enhancement algorithms will lose some low-frequency signals when enhancing high-frequency details, while homomorphic filtering can enhance details while retaining low-frequency signals and can enhance the details based on retaining the original image information.
2. An exposure weighting algorithm based on threshold segmentation and adaptively adjustable Gaussian curve is proposed, which assigns more reasonable weights to well-exposed areas and retains more detailed information.

3. The Laplacian pyramid is improved based on homomorphic filtering, which enhances the edge details of the fused image and generates an image with obvious details.

The rest of this article is organized as follows. In the second part, we discuss the most common and state-of-the-art methods related to multi-exposure image fusion. Section 3 introduces the proposed method in detail. The fourth part analyzes the experimental results for subjective and objective evaluation. The last section discusses the conclusion of this paper and the next steps.

2. Related Works

In recent years, many studies have been conducted on multi-exposure fusion algorithms. Mertens et al. [16] first used contrast, saturation, and well exposure to construct a fused image weight map, then decomposed the source image sequence and weight map into a Laplacian pyramid and a Gaussian pyramid, respectively, and finally fused the resulting image with multi-resolution. The result of this fusion method is very close to reality, and the naked eye can hardly distinguish the large color difference but will lose detail.

Shen et al. [21] proposed a method for exposure fusion using enhanced Laplacian pyramids, the principle of which is to use local weights, global weights, and JND-based saliency weights to estimate the exposure weight map to enhance the detail signal and base signal of the image to construct a novel augmented Laplacian pyramid. This method effectively preserves the texture structure of the image but suffers from artifacts and color distortion due to excessive detail enhancement.

Li et al. [22] proposed extracting image details with a weighted structure tensor, used the Gaussian pyramid of the luminance component of the source image sequence as the guide image, smoothed the weighted Gaussian pyramid of all LDR images using weighted guided image filter (WGIF) [23], and the final result map is obtained by multi-resolution fusion. The detail preservation effect of this method is relatively good; however, the sharpness is flawed where there is a large difference between light and dark between the source image sequences.

Ma et al. [24] proposed a new structural block decomposition method, which first decomposes the source image sequence into three components: signal intensity, signal structure, and average intensity, then fuses each component of different images separately, and finally reconstructs into a fused image. The algorithm is capable of producing images with distinct color appearances but is prone to over-sharpening and local color distortion in areas with large differences in brightness.

Hayat et al. [25] proposed to estimate the initial weights with three indicators of local contrast, brightness, and color dissimilarity where the local contrast was calculated by the dense SIFT [26] descriptor, and then smoothed the weight map with a fast-guided filter, and finally used Pyramid fusion to generate the resulting graph. The algorithm maintains good global contrast but loses some details.

Qi et al. [27] proposed an accurate multi-exposure image fusion method based on low-order features. The principle is to use guided filters to decompose the source image into base layers and detail layers. For the base layer, the method of Ma et al. is used to decompose the image blocks for calculation, and the detail layer is weighted according to the average level of local brightness changes. Finally, the base layer and the detail layer are weighted and fused. This method performs well on image sharpness and color information but is prone to noise and halos.

Huang et al. [28] proposed a multi-exposure image fusion method based on adaptive factor feature evaluation. This method uses the adaptive factor exposure to evaluate the weight, uses the Sobel operator to calculate the texture change weight, and obtains the image by pyramid fusion. The obtained fusion image has better brightness and can retain certain details; however, the details will still be lost in the areas where the image exposure
difference is too large. The detailed method comparison is shown in Table 1. Except for the methods of Ma and Qi, the above fusion methods all use pyramid fusion, and pyramid fusion has the defect of the loss of edge details.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Method</th>
<th>Application</th>
<th>Dataset</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mertens [16]</td>
<td>Contrast, saturation, and well-exposedness are proposed to construct a fused image weight map, fused using Laplacian pyramids.</td>
<td>MATLAB</td>
<td>15 static images by Jacques Joffre, Jesse Levinson, Agrawal [29] and themselves</td>
<td>Subjective comparison test with three algorithms</td>
</tr>
<tr>
<td>Shen [21]</td>
<td>Using local weights, global weights, and JND-based saliency weights to estimate exposure weight maps for enhancing detail and base signals of Laplacian pyramids.</td>
<td>MATLAB</td>
<td>18 static images by themselves</td>
<td>Subjective comparison test with five algorithms</td>
</tr>
<tr>
<td>Li [22]</td>
<td>The image details are extracted with the weighted structure tensor; the Gaussian pyramid of the luminance component of the source image sequence is used as the guide image, the weighted smoothing is applied to all the weighted Gaussian pyramids, and finally the multi-resolution fusion is performed.</td>
<td>MATLAB</td>
<td>6 static images by Laurance Meylan, Dani Lischinski, Jacques Joffre, Martin Cadik, Erik Reinhard, and themselves</td>
<td>Subjective and MEF − SSIM comparative testing of three algorithms</td>
</tr>
<tr>
<td>Ma [24]</td>
<td>It is proposed to decompose the image into three components: signal intensity, signal structure, and average intensity fuse each component separately and finally reconstruct a fused image.</td>
<td>MATLAB</td>
<td>21 static scenes and 19 dynamic scenes by Bartlomiej Okonek, Erik Reinhard, Dani Lischinski, Jianbing Shen, Mertens, Orazio Gallo et al.</td>
<td>Subjective comparison with 12 algorithms, Compare MEF − SSIM with nine algorithms in the static scenario, computational complexity comparison with seven algorithms, average execution time comparison with six algorithms</td>
</tr>
<tr>
<td>Hayat [25]</td>
<td>The local contrast, brightness, and color dissimilar features are proposed to estimate the initial weights, where the local contrast is calculated by DSIFT, and pyramid fusion is used after denoising by a guided filter.</td>
<td>MATLAB</td>
<td>8 images by Mertens, Jianbing Shen et al.</td>
<td>Subjective, $\frac{Q^{AB/F}}{F}$, MEF − SSIM, and NIQE comparative testing of three algorithms</td>
</tr>
<tr>
<td>Qi [27]</td>
<td>The source image is decomposed into a base layer and a detail layer with a guided filter, the base layer is calculated based on image blocks, and the detail layer is weighted according to the local average brightness change.</td>
<td>MATLAB</td>
<td>24 static scenes and 15 dynamic scenes by Ma, Hu, and Sen</td>
<td>The subjectivity of the six algorithms and their mean values $\text{IQA}, \frac{Q^{AB/F}}{F}$, and $\text{MI}$ are compared</td>
</tr>
<tr>
<td>Huang [28]</td>
<td>Based on the Mertens method, the adaptive factor exposure evaluation weight is proposed, the Sobel operator is used to calculate the texture change weight, and finally, the pyramid is fused.</td>
<td>MATLAB</td>
<td>20 static images by Ma</td>
<td>Subjective, MEF − SSIM, and NIQE comparative testing of eight algorithms</td>
</tr>
<tr>
<td>Proposed</td>
<td>Based on the Mertens method, the exposure weight calculated by threshold segmentation and adaptively adjustable Gaussian curve is proposed, and the detail layer of the Laplacian pyramid is enhanced by homomorphic filtering.</td>
<td>MATLAB</td>
<td>17 static images by Ma</td>
<td>Subjective, DPA and $\frac{Q^{AB/F}}{F}$ comparative testing of seven algorithms</td>
</tr>
</tbody>
</table>

Although Shen’s method improves the Laplacian pyramid to enhance the image details, the generated image has the problem of color distortion—the color of the image is quite different from the actual scene. Ma’s results are still poorly exposed where the exposure is poor, and Qi synthesizes methods based on pixel values or building blocks but does not do a good job of removing noise interference on the fusion of detail layers. To produce high-quality images with comfortable visual perception and clear edge details, in this paper, a detail enhancement multi-exposure image fusion algorithm based on homomorphic filtering is proposed based on the Mertens method.
In our method, threshold segmentation and adaptively adjustable Gaussian curve are first used to assign more appropriate weights to well-exposed regions, thereby, preserving more detailed information. Second, the detail layer of the Laplacian pyramid is enhanced by homomorphic filtering, which is used to enhance the edge information and finally to generate an image with realistic colors and rich edge details.

3. Proposed Method

In this paper, a detailed enhancement multi-exposure image fusion algorithm based on homomorphic filtering is proposed. When the average brightness of the image is high, there are more details in the darker areas, and thus a larger weight should be assigned to the low-brightness areas. Similarly, when it is low, a larger weight should be assigned to the place with high brightness. To better preserve image details, this paper proposes an algorithm for calculating exposure weights based on threshold segmentation and an adaptively adjustable Gaussian curve.

Since the multi-resolution fusion algorithm will lose the detailed information of the image, to solve this problem, we propose a detail enhancement algorithm based on homomorphic filtering. The flow of our algorithm is as follows. First, the exposure weight and local contrast weight are calculated to construct an initial weight map, then a fast-guided filter is used to denoise the weight map, and finally, the improved Laplacian pyramid in this paper is performed for multi-resolution fusion. The specific calculation process will be explained in the subsequent subsections, and the schematic diagram of the proposed method is shown in Figure 1. Algorithm 1 shows the detailed calculation process.

![Figure 1. Schematic of the proposed method. The operation process of “∗” in the figure is shown in Equation (17).](image)

3.1. Exposure Weight

The purpose of exposure weighting is to select better exposed areas, which usually contain more image information. The principle of the exposure weight algorithm in the method of Mertens et al. [16] is to use a Gaussian curve to assign the exposure weight after the image is normalized, as shown in Figure 2. The gray value close to 0.5 is considered to be medium exposure and is given a larger weight. When it deviates from 0.5, the larger the deviation, the smaller the weight is given. The specific formula is as follows:

$$W_{in}(i, j) = e^{-\frac{(I_{gray}^{n}(i, j) - 0.5)^2}{2\sigma^2}}$$

(1)

where $I_{gray}^{n}(i, j)$ represents the pixel value in the $i$th row and $j$th column of the normalized grayscale image of the $n$th input image, and $W_{in}(i, j)$ represents the exposure weight value of the $r$th image in the $i$th row and $j$th column, the value range of $n$ is 1, 2, …, $N$, $N$ refers to the number of input image sequences, and $\sigma$ controls the amplitude of the curve and generally takes a value of 0.2.
The method used by Mertens et al. [16] is suitable for images with moderate overall brightness, and this algorithm is not optimal for images with too high or low overall brightness. When the average brightness of the image is high, there are more details in the darker areas, and thus a larger weight should be assigned to the low-brightness areas. Similarly, when it is low, a larger weight should be assigned to the high-brightness areas. To this end, this paper proposes an algorithm for calculating exposure weights based on threshold segmentation and an adaptively adjustable Gaussian curve, which can adaptively adjust the weights according to the average brightness of the input image sequence. The algorithm is divided into two steps: threshold segmentation and the calculation of adaptive exposure weights.

First, we perform threshold segmentation and normalize the image sequence gray image $\hat{I}_n^{gray}$ to the $[0, 1]$ interval. Since the gray value of the darker part of a well-exposed gray image is low, the color is lighter Part of the gray value is high but also contains detailed information. To avoid mistaking the dark and light parts as bad exposure, we adopt an iterative threshold algorithm [30] that divides each image into dark and light parts with a threshold and calculates the exposure weight separately. The specific algorithms of the threshold are as follows:

$$mean\{A(i,j)\} = \frac{1}{r \times c} \sum_{i=1}^{r} \sum_{j=1}^{c} A(i,j)$$  \hspace{1cm} (2)
\[ T_2 = \frac{\text{mean}\left\{ I_{\text{gray}}^n(i,j) > T_1 \right\} + \text{mean}\left\{ I_{\text{gray}}^n(i,j) \leq T_1 \right\}}{2} \]

\[ |T_2 - T_1| < T_0 \]  

where \( \text{mean}\{\cdot\} \) represents the mean value calculation, and the specific algorithm is shown in Equation (2). \( A(i,j) \) represents the area where the mean is calculated, and the size of \( A(i,j) \) is equal to \( r \times c \), the initial value of \( T_1 \) is 0.5, and \( T_0 \) is a very small number. If Equation (3) holds, then \( T_2 \) is the optimal threshold, otherwise, we assign the value of \( T_2 \) to \( T_1 \), and repeat the above steps are iterated until the optimal threshold \( T_2 \) is obtained. Then the image is divided into two parts \( G_1 \) and \( G_2 \), according to the optimal threshold. The \( G_1 \) part is composed of pixels whose gray value is greater than \( T_2 \), and the \( G_2 \) part is composed of pixels whose gray value is less than or equal to \( T_2 \).

Finally, the weight of adaptive exposure is calculated. If the brightness value of this image is low in the whole image sequence, then the part with high brightness is well-exposed, and the weight of this part needs to be appropriately increased. Similarly, if the brightness value of this image is high in the whole image sequence, it is necessary to appropriately increase the weight of the low brightness part. Therefore, we constructed two adaptive variables \( \alpha_n \) and \( \beta_n \), which can reflect the offset between the exposure of the input image and 0.5. The calculation formulas are as follows:

\[ \alpha_n = \sum_{n=1}^{N} \text{mean}\left\{ I_{\text{gray}}^n(i,j) > T_{2n} \right\} - \text{mean}\left\{ I_{\text{gray}}^n(i,j) > T_{2n} \right\} \]  

\[ \beta_n = \sum_{n=1}^{N} \text{mean}\left\{ I_{\text{gray}}^n(i,j) \leq T_{2n} \right\} - \text{mean}\left\{ I_{\text{gray}}^n(i,j) \leq T_{2n} \right\} \]

where \( T_{2n} \) is the optimal threshold of the \( n \)th image; \( \alpha_n \) and \( \beta_n \) represent the adaptive variables of the corresponding parts of \( G_1 \) and \( G_2 \), respectively; and the Gaussian curve is used to assign the exposure weight \( W_{1n} \). The formulas are as follows:

\[ W_{1n}(i_1, j_1) = e^{-\frac{(I_{\text{gray}}^n(i_1,j_1) - (0.5 + \alpha_n))^2}{2\sigma^2}} \]

\[ W_{1n}(i_2, j_2) = e^{-\frac{(I_{\text{gray}}^n(i_2,j_2) - (0.5 + \beta_n))^2}{2\sigma^2}} \]

where \( I_{\text{gray}}^n(i_1,j_1) \in \{ I_{\text{gray}}^n(i,j) > T_{2n} \} \), \( I_{\text{gray}}^n(i_2,j_2) \in \{ I_{\text{gray}}^n(i,j) \leq T_{2n} \} \), \( \sigma \) controls the amplitude of the curve, we take the value as 0.2, and the process of calculating the exposure weight map is shown in Figure 3.

### 3.2. Local Contrast Weight

The computation of local contrast can be used to preserve important details, such as edges and textures. These edge and texture information is included in the gradient change, and thus the Laplacian filter with good edge detection is used to calculate the local contrast weight, and the Laplacian filter is used for each grayscale image. The specific algorithm is as follows:

\[ A_n(i,j) = \left| I_{\text{gray}}^n(i,j) * h(i,j) \right| \]

where \( A_n(i,j) \) represents the local contrast value of the \( n \)th input image in the \( i \)th row and \( j \)th column, \( \left| \cdot \right| \) represents the calculation of the absolute value, \( I_{\text{gray}}^n(i,j) \) represents the pixel value of the \( n \)th grayscale image in the \( i \)th row and \( j \)th column, \( * \) represents the convolution operation, \( h \) is the Laplacian filter kernel, and the value of \( h \) is as follows:

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]
We use the maximum value of the same pixel position in all images as the local contrast weight $W_{2n}(i, j)$, and the specific algorithm is as follows:

$$W_{2n}(i, j) = \begin{cases} 1 & A_n(i, j) = \max \{A_n(i, j), n = 1, 2, \ldots, N \} \\ 0 & \text{otherwise} \end{cases}$$

(11)

where $W_{2n}(i, j)$ represents the local contrast weight value of the $n$th image in the $i$th row and $j$th column.

![Figure 3. Process diagram of exposure weight map calculation.](image)

3.3. Pyramid Fusion Based on Homomorphic Filter Detail Enhancement

The initial weight map is constructed from the calculated two indicators. The obtained initial weight map is noisy and discontinuous. Therefore, it is very important to refine the initial weight map. The fast-guided filter uses the input image as the guide image, and the weight map is refined without damaging the edges, which can remove noise very well. Therefore, we choose to use the guided filter to refine the weight map and normalize it, the specific algorithms are as follows:

$$\hat{W}_n(i, j) = W_{1n}(i, j) \times W_{2n}(i, j)$$

(12)

$$W_n(i, j) = GF_{r, \varepsilon}(\hat{W}_n(i, j), \hat{W}_n(i, j))$$

(13)

where $GF_{r, \varepsilon}(I, G)$ represents the fast-guided filtering, $r$ represents the filter radius, $\varepsilon$ manages the blur degree of the filtering, $I$ represents the input image, and $G$ represents the guided image. The fast bootstrap filter parameter settings are the same as the algorithm of Hayat et al. [25].

Direct weighted fusion can lead to the problem of seams and blurring in the output image, which can be solved by pyramid-based multi-resolution methods; however, pyramid fusion will lose some edge texture details. In order to preserve the detailed information of the image, a pyramid fusion based on homomorphic filter detail enhancement is proposed. Many detail enhancement algorithms lose some low-frequency signals when enhancing high-frequency details, while homomorphic filtering [31] can preserve low-frequency signals while enhancing details; therefore, we use homomorphic filtering for detail enhancement.
The principle of the method is to decompose the weight map and the input image with the Gaussian pyramid and the Laplacian pyramid, respectively. The Laplacian pyramid decomposition decomposes the input image into the base layer and the detail layer. The highest layer is the base layer, and the other layers are the details. We augment each detail layer with homomorphic filtering. For the detailed calculation process of homomorphic filtering, see Algorithm 2 and see Equations (14)–(16) for details:

\[
L = \text{floor}(\log_2 \min(r, c)) - 2 
\]

\[
L\{I_n(i, j)\}^{(l)} = \begin{cases} 
I_n^{(l)}(i, j) - \text{upsample}\left(I_n^{(l+1)}(i, j)\right), & l = 1, 2, \cdots, L - 1 \\
L\{I_n(i, j)\}^{(l)} = I_n^{(l)}(i, j), & l = L 
\end{cases} 
\]

\[
L\{I_n(i, j)\}^{(l)} = \text{homomorphic}\left(L\{I_n(i, j)\}^{(l)}\right), & l = 1, 2, \cdots, L - 1 
\]

where floor(·) means rounding towards negative infinity, r and c are the height and width of the input image, respectively; min(·) means taking the minimum value function; \(I_n^{(l)}(i, j)\) means the \(l\)th layer of the \(n\)th layer for pixel values in the \(i\)th row and \(j\)th column; upsample(·) is an up-sampling operation; \(L\{\cdot\}^{(l)}\) represents the Laplacian pyramid image of the \(l\)th layer; and homomorphic(·) represents the Homomorphic filtering operation. After the enhanced Laplacian pyramid is obtained, it is fused and reconstructed with the Gaussian pyramid of the weight map to obtain the final fused image. The equations are as follows:

\[
L\{F(i, j)\}^{(l)} = \sum_{n=1}^{N} G\{W_n(i, j)\}^{(l)} \times L\{I_n(i, j)\}^{(l)} 
\]

\[
L\{F(i, j)\}^{(L-1)} = L\{F(i, j)\}^{(L-1)} + \text{upsample}\left(L\{F(i, j)\}^{(L-1+1)}\right), & l = 1, 2, \cdots, L - 1 
\]

\[
F(i, j) = L\{F(i, j)\}^{(1)} 
\]

where \(G\{\cdot\}^{(l)}\) represents the Gaussian pyramid image of the \(n\)th layer, and \(F(i, j)\) is the pixel value of the fusion image in the \(i\)th row and the \(j\)th column—that is, the final output image.

**Algorithm 2: Homomorphic filtering algorithm.**

Parameter: \(n0.1, D0.0, r0.7, r0.1\)

Input:
Laplacian pyramid of the source image \(L\{I_n\}^{(l)}, 1 \leq l \leq L - 1\)

Output: Laplacian pyramid with enhanced detail \(L\{I_n\}^{(l)}\)

1: for each layer of \(L\{I_n\}^{(l)}\) do
2: \(L\{I_n\}^{(l)} := \text{log}\left(L\{I_n\}^{(l)} + 1\right)\)
3: The Fourier transform \(L_{fft}\{I_n\}^{(l)} := \text{fft2}\left(L\{I_n\}^{(l)}\right)\)
4: \(D := \text{sqrt}(t^2 + \beta^2)\)
5: \(H := r_0 \left(1 + (D_0/D_1)^{2^m}\right)\)
6: Filtering the Fourier transform \(L_{fft}\{I_n\}^{(l)} := L_{fft}\{I_n\}^{(l)} \ast H\)
7: Inverse Fourier transform \(L\{I_n\}^{(l)} := \text{ifft2}\left(L_{fft}\{I_n\}^{(l)}\right)\)
8: Indexation \(L\{I_n\}^{(l)} := \exp\left(L\{I_n\}^{(l)}\right) - 1\)
9: Take the real part to get the enhanced pyramid \(L\{I_n\}^{(l)} := \text{real}\left(L\{I_n\}^{(l)}\right)\)
10: end for
4. Experimental Results and Analysis

In this section, to examine the performance of the proposed fusion method, seventeen different LDR multiple-exposure image sequences [24] were selected (which include different scenes) as listed in Table 2. We tested our method using these seventeen natural scenes with different exposure levels and compared them with the seven most popular existing algorithms [16,21,22,24,25,27,28]. Four representative sets of input images were selected for presentation in Figure 4, and Figures 5–8 show the results of these four sets of images fused with different methods. All experiments were run with MATLAB R2019a on a PC with an Intel i5 6200U @ 2.40 GHz processor and 4.00 GB RAM.

Table 2. Information about the static source sequences.

<table>
<thead>
<tr>
<th>Source Sequence</th>
<th>Size</th>
<th>Image Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arno</td>
<td>339 × 512 × 3</td>
<td>Bartłomiej Okonek [32]</td>
</tr>
<tr>
<td>Cave</td>
<td>512 × 384 × 4</td>
<td>Bartłomiej Okonek [32]</td>
</tr>
<tr>
<td>Chinese Garden</td>
<td>512 × 340 × 3</td>
<td>Bartłomiej Okonek [32]</td>
</tr>
<tr>
<td>Church</td>
<td>335 × 512 × 3</td>
<td>Jianbing Shen [21]</td>
</tr>
<tr>
<td>Farmhouse</td>
<td>512 × 341 × 4</td>
<td>HDR projects [33]</td>
</tr>
<tr>
<td>house</td>
<td>512 × 340 × 4</td>
<td>Mertens [16]</td>
</tr>
<tr>
<td>Kluki</td>
<td>512 × 341 × 3</td>
<td>Bartłomiej Okonek [32]</td>
</tr>
<tr>
<td>Lamp</td>
<td>512 × 384 × 15</td>
<td>Martin Cadik [34]</td>
</tr>
<tr>
<td>Landscape</td>
<td>512 × 341 × 3</td>
<td>HDRsoft [35]</td>
</tr>
<tr>
<td>Laurenziana</td>
<td>356 × 512 × 3</td>
<td>Bartłomiej Okonek [32]</td>
</tr>
<tr>
<td>Madison Capitol</td>
<td>512 × 384 × 30</td>
<td>Chaman Singh Verma [36]</td>
</tr>
<tr>
<td>Mask</td>
<td>512 × 341 × 3</td>
<td>HDRsoft [35]</td>
</tr>
<tr>
<td>Ostrow</td>
<td>341 × 512 × 3</td>
<td>Bartłomiej Okonek [32]</td>
</tr>
<tr>
<td>Room</td>
<td>512 × 340 × 3</td>
<td>Pangeasoft [37]</td>
</tr>
<tr>
<td>Studio</td>
<td>512 × 341 × 5</td>
<td>HDRsoft [35]</td>
</tr>
<tr>
<td>Tower</td>
<td>512 × 341 × 3</td>
<td>Jacques Joffre [35]</td>
</tr>
<tr>
<td>Window</td>
<td>384 × 512 × 3</td>
<td>Hvdwolf [38]</td>
</tr>
</tbody>
</table>

Figure 4. Four sets of multiple exposure image sequences. (a) ‘Cave’ image sequence. (b) ‘Lamp’ image sequence. (c) ‘House’ image sequence. (d) ‘Tower’ image sequence.
Figure 4. Four sets of multiple exposure image sequences. (a) ‘Cave’ image sequence. (b) ‘Lamp’ image sequence. (c) ‘House’ image sequence. (d) ‘Tower’ image sequence.

Figure 5. Fusion results for Figure 4a. (a) Mertens et al.’s method [16]. (b) Shen et al.’s method [21]. (c) Li et al.’s method [22]. (d) Ma et al.’s method [24]. (e) Hayat et al.’s method [25]. (f) Qi et al.’s method [27]. (g) Huang et al.’s method [28]. (h) The proposed method.

Figure 6. Fusion results for Figure 4b. (a) Mertens et al.’s method [16]. (b) Shen et al.’s method [21]. (c) Li et al.’s method [22]. (d) Ma et al.’s method [24]. (e) Hayat et al.’s method [25]. (f) Qi et al.’s method [27]. (g) Huang et al.’s method [28]. (h) The proposed method.
Figure 7. Fusion results for Figure 4c. (a) Mertens et al.’s method [16]. (b) Shen et al.’s method [21]. (c) Li et al.’s method [22]. (d) Ma et al.’s method [24]. (e) Hayat et al.’s method [25]. (f) Qi et al.’s method [27]. (g) Huang et al.’s method [28]. (h) The proposed method.

Figure 8. Fusion results for Figure 4d. (a) Mertens et al.’s method [16]. (b) Shen et al.’s method [21]. (c) Li et al.’s method [22]. (d) Ma et al.’s method [24]. (e) Hayat et al.’s method [25]. (f) Qi et al.’s method [27]. (g) Huang et al.’s method [28]. (h) The proposed method.
4.1. Subjective Analysis

Figures 5–8 show the overall results and partial magnified images of the four experiments. The method of Mertens [16] has adequate color vibrancy; however, there is a certain loss of details, as shown in Figures 7a and 8a. The method of Shen [21] has rich texture details; however, there are artifacts at the junction of light and dark and serious color distortion of the entire image due to excessive detail enhancement. For example, there are obvious black shadows at the junction of light and dark at the entrance of the cave in Figure 5b, and the color of the lamp is severely distorted in Figure 6b.

The method of Li [22] uses a weighted structure tensor to extract details for detail enhancement; however, the brightness and clarity are not good in some places, for instance the branches and leaves in Figure 7c are missing. In Ma’s method [24], based on the structural block decomposition, the image block is decomposed into the signal intensity, signal structure and average intensity, which exhibit good global contrast but are prone to over-sharpening, resulting in local color distortion, as shown in Figures 5d, 6d and 7d.

Hayat [25] used the dense SIFT descriptor to calculate the contrast and the smoothing weight of the guided filter. The generated image maintains a good global contrast; however, it is easy to lose details. As shown in Figure 6e, the exposure of the desk lamp is too high, and in Figure 8e, the texture details of the cloud are lost and the clarity is poor.

The method of Qi [27] performs well in image clarity and color saturation but is prone to noise and halos at the connection between light and dark. For example, there are orange noise spots at the entrance of the cave in Figure 5f, there are obvious noise spots on the surface of the desk lamp in Figure 6f, and the leaves have unnatural halos in Figure 7f. The method of Huang [28] has moderate brightness and can retain certain details, however, still loses details in areas where the image exposure difference is too large. For example, the leaves and branches in Figure 7g are missing a part.

The algorithm proposed in this paper shows advantages in all aspects. As shown in Figure 5h, the light–dark transition area of the cave entrance is rich in details, and the color is more realistic. The edge and texture of the lamp in Figure 6h can be seen, and the leaves and branches are visible in Figure 7h. The detailed texture of the cloud is richer without distortion in Figure 8h. To summarize, compared with the other seven methods, our method can preserve more details and edge information and exhibits a comfortable visual effect.

4.2. Objective Evaluation

The evaluation metric can objectively display the comprehensive performance of the algorithm. We used two objective evaluation indicators to evaluate. Detail-preserving assessment (DPA) [39] is used to evaluate detail preservation, and \( Q^{AB/F} \) [40] utilizes gradients to measure the edge information between fused and source images. The DPA and \( Q^{AB/F} \) values of seventeen different groups of fused images were obtained through eight different algorithms, as shown in Table 3, and the bolded ones are the maximum DPA and \( Q^{AB/F} \) values using different algorithms.

The higher the DPA score, the higher the detail retention rate. DPA is defined as follows:

\[
DPA(r,c) = -\log \left( \frac{\sum_{x=1}^{r} \sum_{y=1}^{c} \left| E^{R}(x,y) - \max(E^{A}(x,y)) \right|}{r \times c} \right)
\]  

where \( E^{R}(x,y) \) is the exposure weight of the resulting image in row \( x \) and column \( y \), \( \max(E^{A}(x,y)) \) is the maximum exposure weight of the input image in row and column, and the image size is equal to \( r \times c \).
The DPA score is calculated by comparing the resulting image to the input images; thus, when the resulting image has more detail than all the input images, the numerator in the above equation is larger, and the DPA value becomes low. Table 3 shows the numerical results of the method in this paper and seven other popular methods. Among the seventeen comparison charts, there are ten DPA test results of the algorithm in this paper with the maximum value. For the remaining seven DPA test results, the algorithm in this paper ranks second; thus, the detail retention ability of the algorithm in this paper is excellent.

The larger the $Q^{AB/F}$ value, the better the ability to maintain edge information. As shown in Table 3, among the seventeen sets of comparison results, the $Q^{AB/F}$ value of the algorithm in this paper is the largest in the twelve groups. The remaining five groups are also good, and, in general, this paper has better edge preservation performance.

As shown in Figure 9, in the mean test results of the method proposed in this paper, DPA ranks second, and $Q^{AB/F}$ ranks first. The first place of DPA means is the method proposed by Shen et al. [21]; however, the $Q^{AB/F}$ mean of their method ranks last, indicating that its ability to preserve edge information is not very superior. According to the line graph of the average value of DPA and $Q^{AB/F}$ in Figure 9, it can be seen that the method proposed in this paper ranks first. In general, the image fusion method proposed in this paper can better preserve the edge information of the source image compared with the tested methods and can retain most of the details of the image.

### Table 3. Test results of DPA and $Q^{AB/F}$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Arno</td>
<td>1.745/0.616</td>
<td>2.105/0.369</td>
<td>1.746/0.57</td>
<td>1.476/0.593</td>
<td>1.63/0.573</td>
<td>1.462/0.579</td>
<td>1.689/0.629</td>
<td>2.296/0.622</td>
</tr>
<tr>
<td>Cave</td>
<td>1.735/0.795</td>
<td>2.249/0.561</td>
<td>2.26/0.834</td>
<td>1.905/0.741</td>
<td>1.797/0.801</td>
<td>1.915/0.798</td>
<td>1.716/0.719</td>
<td>2.351/0.822</td>
</tr>
<tr>
<td>Chinese Garden</td>
<td>1.427/0.812</td>
<td>2.173/0.475</td>
<td>1.628/0.825</td>
<td>1.135/0.826</td>
<td>1.455/0.822</td>
<td>1.195/0.812</td>
<td>1.335/0.825</td>
<td>2.07/0.831</td>
</tr>
<tr>
<td>Church</td>
<td>1.788/0.846</td>
<td>1.823/0.634</td>
<td>2.051/0.852</td>
<td>1.817/0.852</td>
<td>2.051/0.844</td>
<td>1.742/0.829</td>
<td>1.62/0.835</td>
<td>2.125/0.871</td>
</tr>
<tr>
<td>Farmhouse</td>
<td>2.572/0.8</td>
<td>2.481/0.645</td>
<td>2.544/0.804</td>
<td>2.357/0.804</td>
<td>2.622/0.804</td>
<td>2.39/0.804</td>
<td>2.39/0.804</td>
<td>2.76/0.823</td>
</tr>
<tr>
<td>House</td>
<td>1.172/0.693</td>
<td>1.821/0.407</td>
<td>1.422/0.698</td>
<td>1.131/0.567</td>
<td>1.296/0.685</td>
<td>1.257/0.681</td>
<td>1.152/0.707</td>
<td>1.496/0.712</td>
</tr>
<tr>
<td>Kluki</td>
<td>1.433/0.823</td>
<td>1.821/0.335</td>
<td>1.662/0.837</td>
<td>1.386/0.83</td>
<td>1.57/0.831</td>
<td>1.37/0.811</td>
<td>1.404/0.839</td>
<td>1.879/0.837</td>
</tr>
<tr>
<td>Lamp</td>
<td>1.235/0.743</td>
<td>2.797/0.378</td>
<td>1.536/0.73</td>
<td>0.892/0.724</td>
<td>1.168/0.708</td>
<td>0.869/0.721</td>
<td>1.118/0.749</td>
<td>1.417/0.731</td>
</tr>
<tr>
<td>Landscape</td>
<td>1.81/0.611</td>
<td>2.281/0.339</td>
<td>2.012/0.614</td>
<td>1.211/0.631</td>
<td>1.886/0.604</td>
<td>1.197/0.594</td>
<td>1.609/0.641</td>
<td>2.534/0.626</td>
</tr>
<tr>
<td>Laurenziana</td>
<td>1.416/0.807</td>
<td>1.905/0.572</td>
<td>1.551/0.82</td>
<td>1.12/0.814</td>
<td>1.459/0.816</td>
<td>1.053/0.799</td>
<td>1.417/0.818</td>
<td>1.989/0.826</td>
</tr>
<tr>
<td>Madison Capitol</td>
<td>1.02/0.817</td>
<td>2.384/0.443</td>
<td>1.082/0.819</td>
<td>0.838/0.781</td>
<td>1.046/0.787</td>
<td>0.83/0.766</td>
<td>0.964/0.822</td>
<td>1.527/0.816</td>
</tr>
<tr>
<td>Mask</td>
<td>1.631/0.828</td>
<td>2.443/0.496</td>
<td>2.084/0.847</td>
<td>1.26/0.841</td>
<td>1.729/0.846</td>
<td>1.247/0.822</td>
<td>1.51/0.835</td>
<td>2.357/0.853</td>
</tr>
<tr>
<td>Ostrow</td>
<td>2.139/0.6</td>
<td>2.035/0.37</td>
<td>2.17/0.539</td>
<td>1.863/0.569</td>
<td>2.136/0.545</td>
<td>1.869/0.576</td>
<td>2.102/0.606</td>
<td>2.337/0.641</td>
</tr>
<tr>
<td>Room</td>
<td>1.968/0.789</td>
<td>2.376/0.478</td>
<td>2.11/0.815</td>
<td>1.833/0.806</td>
<td>1.874/0.808</td>
<td>1.839/0.792</td>
<td>1.956/0.798</td>
<td>2.348/0.817</td>
</tr>
<tr>
<td>Studio</td>
<td>1.334/0.735</td>
<td>2.475/0.49</td>
<td>1.632/0.77</td>
<td>1.046/0.709</td>
<td>1.23/0.732</td>
<td>1.095/0.724</td>
<td>1.269/0.743</td>
<td>1.923/0.773</td>
</tr>
<tr>
<td>Tower</td>
<td>1.78/0.781</td>
<td>2.316/0.504</td>
<td>2.23/0.811</td>
<td>1.467/0.81</td>
<td>1.932/0.81</td>
<td>1.442/0.787</td>
<td>1.683/0.782</td>
<td>2.56/0.819</td>
</tr>
<tr>
<td>Window</td>
<td>1.775/0.779</td>
<td>1.989/0.569</td>
<td>2.066/0.793</td>
<td>1.707/0.787</td>
<td>1.899/0.792</td>
<td>1.803/0.772</td>
<td>1.665/0.768</td>
<td>2.196/0.804</td>
</tr>
<tr>
<td>Average</td>
<td>1.646/0.757</td>
<td>2.19/0.486</td>
<td>1.87/0.764</td>
<td>1.439/0.746</td>
<td>1.693/0.753</td>
<td>1.438/0.744</td>
<td>1.564/0.761</td>
<td>2.127/0.778</td>
</tr>
</tbody>
</table>

**Figure 9.** Use different methods to analyze the mean values of $DPA$, $Q^{AB/F}$, and both of them.

### 4.3. The Comparative Experiment of Adaptive Exposure Weight Calculation

We conducted experiments with 10 images with and without our adaptive exposure weighting algorithm to show the effectiveness of the algorithm. We only replaced our exposure weighting algorithm with the Mertens calculation exposure weighting algorithm,
and the other calculation steps remained unchanged. Figure 10 shows $Q^{AB/F}$ comparison of the proposed and alternative exposure weighting algorithms. It can be seen that the $Q^{AB/F}$ value of the adaptive exposure weight calculation algorithm we used was higher than that of the other algorithm, that is to say, our adaptive exposure weight algorithm had a better ability to maintain edge information.

![Figure 10. Q^{AB/F} comparison of the proposed and alternative exposure weighting algorithms.](image)

5. Conclusions

In this paper, we proposed a multi-exposure image fusion method for detail enhancement based on homomorphic filtering. We noticed that high-exposure images have more details at low exposure values, and similarly, low-exposure images have more details at high exposure values. Therefore, we proposed an exposure-weighting algorithm based on threshold segmentation and an adaptively adjustable Gaussian curve. The initial weight map was constructed by using exposure weight and local contrast weight, and then the initial weight map was denoised and refined by fast-guided filtering to obtain a comprehensive weight map.

Finally, the input image was decomposed into the improved Laplacian pyramid. We conducted the decomposition of comprehensive weights into a Gaussian pyramid and the multi-resolution fusion of pyramids into a result map. We compared seventeen sets of static natural image sequences with different exposure levels and then compared and analyzed the algorithms from both subjective and objective aspects. The experimental results show that the method proposed in this paper can better preserve the edge details of the source image and generate images with uniform illumination distribution. At present, the algorithm in this paper is only applicable in static scenes and cannot solve the ghosting problem of dynamic image fusion. In the future, we plan to study the problem of ghosting removal in dynamic scenes to enhance the practicability of the algorithm.

**Author Contributions:** Methodology, Y.H.; software, C.X.; validation, Z.L., F.L., B.F. and L.C.; resources, C.N.; data curation, D.W.; writing—original draft preparation, Y.H.; writing—review and editing, C.X.; visualization, Y.H.; supervision, C.X.; project administration, C.X.; and funding acquisition, C.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key Research and Development Program of China (No. 2019YFC0117800).

**Acknowledgments:** Input image sequences were provided by Kede Ma, Department of Computer Science, City University of Hong Kong.

**Conflicts of Interest:** The authors declare no conflict of interest.


